



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

Kierunek studiów: Informatyka stosowana

Praca dyplomowa – magisterska

**ESTYMACJA PARAMETRÓW MODELU  
WIELOAGENTOWEGO ROZPRZESTRZENIANIA SIĘ  
KORONAWIRUSA ZE STRUKTURĄ WIEKU**

Jakub Jaszczyk

słowa kluczowe:  
COVID-19, estymacja parametrów, optymalizacja, modelowanie agentowe, model SEIR

krótkie streszczenie:

W niniejszej pracy przedstawiono sposób estymacji parametrów epidemii COVID-19 przy użyciu modelu agentowego. W celu estymacji wykorzystano podejście oparte o optymalizację wartości funkcji błędu. W ramach badań porównano kilka różnych algorytmów optymalizacji oraz różne podejścia. Wyniki estymacji częściowo pokrywają się z parametrami medycznymi, co świadczy o skuteczności proponowanego podejścia do estymacji.

Opiekun pracy dyplomowej	dr inż. Jarosław Drapała		
	Tytuł/stopień naukowy/imię i nazwisko		
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego			
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:\**

- a) kategorii A (akta wieczyste)
- b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

*\* niepotrzebne skreślić*

pieczęć wydziałowa

Wrocław, rok 2021



## SPIS TREŚCI

<b>Streszczenie</b> . . . . .	2
<b>1. Wstęp</b> . . . . .	3
1.1. Modelowanie COVID-19 . . . . .	3
1.2. Estymacja parametrów . . . . .	3
1.3. Cel i zakres pracy . . . . .	4
1.4. Niepewność pomiarów . . . . .	4
1.5. Przegląd rozdziałów . . . . .	5
<b>2. Modelowanie rozprzestrzeniania się epidemii</b> . . . . .	6
2.1. Sposoby modelowania . . . . .	6
2.2. Modele epidemiologiczne . . . . .	9
2.3. Pakiety do modelowania agentowego . . . . .	11
2.4. Opis środowiska Mesa . . . . .	13
<b>3. Analiza proponowanego rozwiązania</b> . . . . .	16
3.1. Parametry COVID-19 . . . . .	16
3.2. Źródła danych . . . . .	20
3.3. Istniejące rozwiązania . . . . .	20
3.4. Opracowane rozwiązanie . . . . .	22
3.5. Plan badań symulacyjnych . . . . .	24
<b>4. Badania symulacyjne</b> . . . . .	31
4.1. Badania wstępne . . . . .	32
4.2. Ewaluacja i weryfikacja otrzymanych wartości . . . . .	38
<b>Zakończenie</b> . . . . .	40
Perspektywy rozwojowe . . . . .	40
Podsumowanie . . . . .	40
<b>Bibliografia</b> . . . . .	42
<b>Spis rysunków</b> . . . . .	45

## **STRESZCZENIE**

W niniejszej pracy przedstawiono sposób estymacji parametrów epidemii COVID-19 przy użyciu modelu agentowego. Model agentowy umożliwia prostsze i bardziej realistyczne odwzorowanie skomplikowanego systemu w porównaniu do modelu matematycznego. Na podstawie analizy literatury przedstawiony został model epidemii COVID-19, będący rozszerzeniem klasycznego modelu SEIR. Ponadto zebrano informacje dotyczące parametrów medycznych oraz omówiono inne prace o tematyce estymacji. W celu estymacji wykorzystano podejście oparte o optymalizację wartości funkcji błędu. W ramach badań porównano kilka różnych algorytmów optymalizacji oraz różne podejścia. Otrzymane wartości parametrów zostały porównane z wartościami medycznymi dla kilku różnych zbiorów danych. Wyniki estymacji częściowo pokrywają się z parametrami medycznymi, co świadczy o skuteczności proponowanego podejścia do estymacji. Przedstawiony model można rozszerzać i sprawdzać inne metody estymacji parametrów. Istotnym czynnikiem okazał się również dobór danych dotyczących pandemii. Niniejsza praca stanowi skuteczny przegląd kilku metod estymacji, optymalizacji oraz samej epidemii COVID-19.

## **ABSTRACT**

This paper presents a method of estimating the parameters of the COVID-19 epidemic using the agent-based model. The agent-based model allows a simpler and more realistic representation of a complex system compared to the mathematical model. Based on the analysis of the literature, the COVID-19 epidemic model was presented, which is an extension of the classic SEIR model. In addition, information on medical parameters was collected and other works on estimation were discussed. In order to estimate, an approach based on optimization of the error function value was used. The study compared several different optimization algorithms and different approaches. The obtained parameter values were compared with the medical values for several different data sets. The estimation results partially overlap with the medical parameters, which proves the effectiveness of the proposed estimation approach. The presented model can be extended and other methods of parameter estimation can be tested. The selection of pandemic data also proved to be an important factor. This paper provides an effective overview of several estimation and optimization methods as well as the COVID-19 epidemic itself.

## 1. WSTĘP

### 1.1. MODELOWANIE COVID-19

Zagadnienie modelowania epidemii jest już poruszane od dawna i obecnie najbardziej podstawowy model składa się z trzech stanów w jakich może się znajdować dana osoba. Są to stany: podatny na zarażenie, zarażony, odporny. Od pierwszych liter angielskich nazw tych stanów model ten nazywany jest SIR. Tak prosty model można rozszerzać na wiele sposobów, dodawać nowe stany. Przykładem bardziej zaawansowanego podejścia jest model SEIR. Zastępuje on stan zarażony dwoma nowymi stanami. Stan zarażony, ale nie mogący zarażać innych oraz stan zarażony i mogący zarażać. Daje to łącznie cztery możliwe stany w jakich mogą znajdować się poszczególne osobniki w populacji. Takie podejście pozwala na lepsze oddanie rzeczywistego rozwoju wielu chorób. Przejścia pomiędzy kolejnymi stanami są opisywane współczynnikami liczbowymi z zakresu od 0 do 1. Parametr  $\beta$  odpowiada za przejście  $S \rightarrow E$  i oznacza prawdopodobieństwo zarażenia się w wyniku kontaktu z osobą zarażoną. Odwrotność  $\sigma$  jest średnią liczbą dni potrzebną do przejścia  $E \rightarrow I$ . Odwrotność parametru  $\gamma$  oznacza średnią liczbę dni potrzebną do wyzdrowienia, czyli przejścia  $I \rightarrow R$ .

W zagadnieniach modelowania również można wyróżnić wiele podejść. Najczęściej spotykane i mające najsolidniejsze podstawy są modele matematyczne. Do tej klasy można zaliczyć bardzo wiele podejść, od prostego modelu liniowego do skomplikowanych rozwiązań stochastycznych lub sieci neuronowych. Niestety uchwycenie wszystkich szczegółów przy pomocy równań może być ciężkim zadaniem i z tego powodu można stosować również modele agentowe. Takie modele są oparte o interakcje agentów (bytów, ludzi) ze środowiskiem. Za ich pomocą można tworzyć skomplikowane i dynamiczne systemy o zaskakujących skutkach wynikających z działań pojedynczych agentów.

### 1.2. ESTYMACJA PARAMETRÓW

Estymacja pozwala na oszacowanie wartości nieznanego parametru. W rozważanym problemie poszukiwanymi parametrami są 3 wartości opisujące współczynniki przejścia pomiędzy kolejnymi stanami modelu SEIR. Model ten jest jednym z podstawowych modeli służących do opisywania rozwoju epidemii. W ramach niniejszej pracy postanowiono zastosować optymalizację funkcji błędu w celu estymacji tych nieznanymi parametrów. Zadaniem optymalizacyjnym jest minimalizowanie wartości funkcji błędu obliczanej z wykorzystaniem danych rzeczywistych oraz danych pochodzących z modelu. Do zagadnienia estymacji można podejść na różne sposoby. Jednym z podejść jest zastosowanie

metod statystycznych, takich jak metody Monte Carlo. Istotny wpływ w nich odgrywa losowość i wielokrotne powtarzanie eksperymentu. Możliwa przestrzeń rozwiązań jest losowo próbkowana. Z wykorzystaniem metod statystycznych rozwiązania uzyskane w ten sposób są łączone w jedno rozwiązanie końcowe. Kolejną możliwością może być podejście do problemu estymacji jako problemu optymalizacji. W takim przypadku można zastosować dowolny optymalizator, który może działać w sposób deterministyczny lub stochastyczny. Poszukiwanymi estymatami są wartości argumentów, dla których optymalizowana funkcja przyjmuje optimum.

### **1.3. CEL I ZAKRES PRACY**

Celem pracy była estymacja parametrów epidemii COVID-19 z wykorzystaniem modelu agentowego oraz metod optymalizacji. Rozważany był model SEIR, w którym pojedynczy agent może znajdować się w danym czasie w jednym z czterech stanów, a przejścia pomiędzy nimi są dokładnie zdefiniowane. W tym celu należało przeanalizować istniejące rozwiązania i dokonać zasadnego rozszerzenia modelu o nowe stany. Na podstawie takich analiz wyodrębniono nowy stan reprezentujący osoby zmarłe. W celu estymacji parametrów postanowiono wykorzystać podejście oparte o optymalizację wartości funkcji błędu. Należało rozpoznać możliwe metody służące do optymalizacji. Wybrane metody zostały wykorzystane w środowisku programistycznym. Zdecydowano się na przeprowadzenie badań mających wyłonić metody najbardziej obiecujące. W końcowej fazie badań postanowiono porównać otrzymane wyniki z danymi pochodzącymi z literatury. Całość pracy podsumowano i wskazano możliwe kierunki rozwoju.

### **1.4. NIEPEWNOŚĆ POMIARÓW**

W modelowaniu rozwoju COVID-19 można wciąż napotkać wiele niepewności, ponieważ pomimo roku trwania pandemii wciąż nie mamy pełnego obrazu dotyczącego jej skutków oraz przebiegu. Jednym z problemów w modelowaniu rozwoju COVID-19 jest niepewność danych. Dane są zbierane i publikowane przez różne instytucje, agregowane w repozytoria i przetwarzane w celu korekty nieprawidłowości. Wciąż jednak są to dane dotyczące osób przebadanych na COVID-19, a przy znacznej liczbie przypadków asymptomatycznych lub ukrytych z innych powodów nie oddają one prawdziwego przebiegu pandemii. Szacuje się, że większość przypadków nie zostaje wykryta. Zależy to też od dostępności testów i strategii danego państwa, które przykładowo może testować na życzenie albo tylko osoby z wyraźnymi objawami. Analizując dane można też zauważyć, że wzrosty liczby zakażeń zbiegają się w czasie ze zwiększeniem liczby testów. Trudne jest w tym przypadku stwierdzenie, co jest przyczyną, a co skutkiem. Czy więcej zarażeń zostaje wykrytych, bo jest więcej testów, czy jest więcej testów, ponieważ więcej osób jest zarażonych? Takie niepewności prowadzą do problemów z modelowaniem tej choroby. W

ramach niniejszej pracy postanowiono modelować dane dotyczące dziennej liczby zarażeń, wyzdrowień oraz zgonów. W tym celu model w trakcie działania nie zwraca licznosci populacji w danym kroku czasowym, jak to ma miejsce w klasycznym podejściu SEIR, a odpowiednie wartości przedstawiające liczbę nowych dziennych przypadków.

## **1.5. PRZEGLĄD ROZDZIAŁÓW**

W pierwszej części zaprezentowano i omówiono sposoby modelowania epidemii z wyróżnieniem kilku najbardziej istotnych modeli opisywanych równaniami różniczkowymi. Opisano podejście oparte o modele matematyczne, agentowe i automaty komórkowe. Krótko zostały też wspomniane możliwe rozwiązania informatyczne służące do modelowania agentowego. W następnej części pracy dokonano przeglądu istniejącej literatury, wskazano źródła danych, zebrano dane pochodzące z analiz medycznych oraz dokładnie opisano wykorzystany model i zaproponowano plan badań symulacyjnych. W dalszej części przedstawiono i przeanalizowano wyniki przeprowadzonych badań, które głównie skupiają się na problemie optymalizacji oraz transformacji danych. W części końcowej podsumowano treści przedstawione w ramach niniejszej pracy.

## 2. MODELOWANIE ROZPRZESTRZENIANIA SIĘ EPIDEMII

W niniejszym rozdziale przedstawiono krótki opis możliwych podejść do modelowania zjawisk zachodzących w otaczającym nas świecie. W dalszej części dokładniej zaprezentowano modele wykorzystywane do modelowania epidemii oraz stojące za nimi równania różniczkowe. Pokróćce omówiono też możliwe pakiety komputerowe, które można wykorzystać do modelowania agentowego. W sposób szczególny postanowiono opisać pakiet Mesa, ponieważ został on wykorzystany w ramach tej pracy.

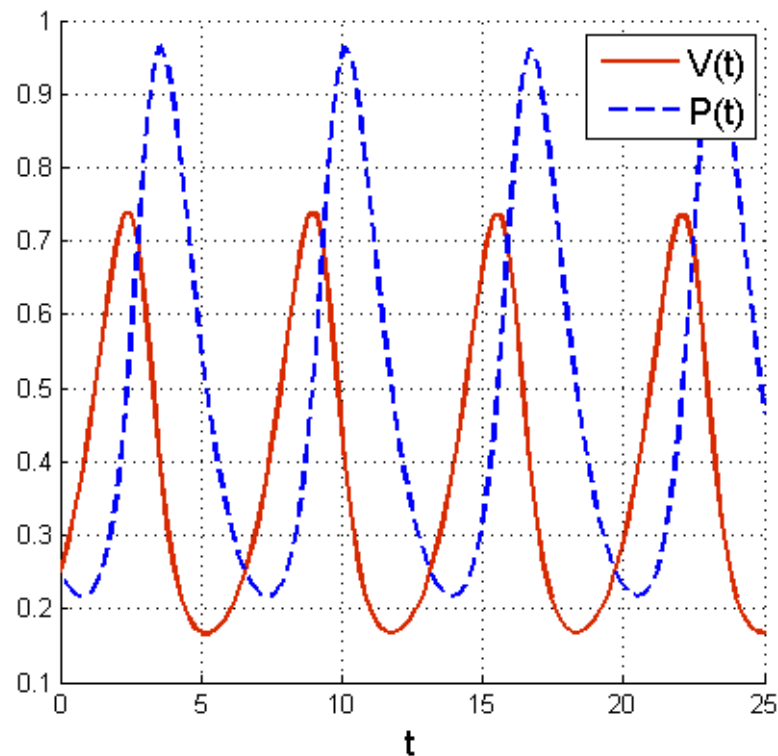
### 2.1. SPOSOBY MODELOWANIA

W modelowaniu matematycznym wykorzystuje się narzędzia, konstrukty matematyczne w celu opisywania zjawisk, relacji, społeczeństw czy ogólnie działania otaczającego nas świata. Modele matematyczne mają bardzo silne fundamenty rozwijane przez ludzkość na przestrzeni wieków. Dodatkową zaletą jest ich opisywalność oraz możliwość udowadniania poprawności poprzez wykorzystanie dostępnego aparatu matematycznego. Prosty modelem tego typu może być funkcja liniowa, ale często można spotkać skomplikowane, wieloetapowe funkcje. Można do nich zaliczyć drzewa decyzyjne, sztuczne sieci neuronowe i inne modele związane z uczeniem maszynowym. Często w celu opisywania zjawisk dynamicznych wykorzystuje się równania różniczkowe. Do klasy takich opisów matematycznych można zaliczyć klasyczne model epidemiologiczne czy przykładowo model Lotki-Volterry [26], który zostanie wykorzystany dalej jako przykład.

Model Lotki-Volterry opisuje system, w którym znajdują się dwie populacje. Są to odpowiednio populacja ofiar i populacja drapieżników. Przykładowy przebieg równań opisujących licznosc obu populacji został przedstawiony na rysunku 2.1, gdzie  $P(t)$  oznacza ofiary, a  $V(t)$  drapieżniki. Jest to prosty układ dwóch równań różniczkowych z czterema parametrami oznaczającymi rozrodczość (przyrost) i śmiertelność (spadek) obu populacji. Model Lotki-Volterry bierze pod uwagę istnienie dwóch populacji, dwóch grup agentów, co doprowadza do możliwości rozpatrzenia podobnego modelu przy pomocy modelowania agentowego.

W modelowaniu agentowym równania nie są wykorzystywane bezpośrednio, a polega się na interakcjach agentów z innymi agentami oraz środowiskiem. Takie podejście zapewnia bardziej wierne odwzorowanie rzeczywistych warunków oraz zachowań zachodzących w wybranych środowiskach, które podlegają symulacji. Ponadto modelowanie agentowe zapewnia dokładniejszą kontrolę nad działaniami indywidualnych jednostek, a takie działa-





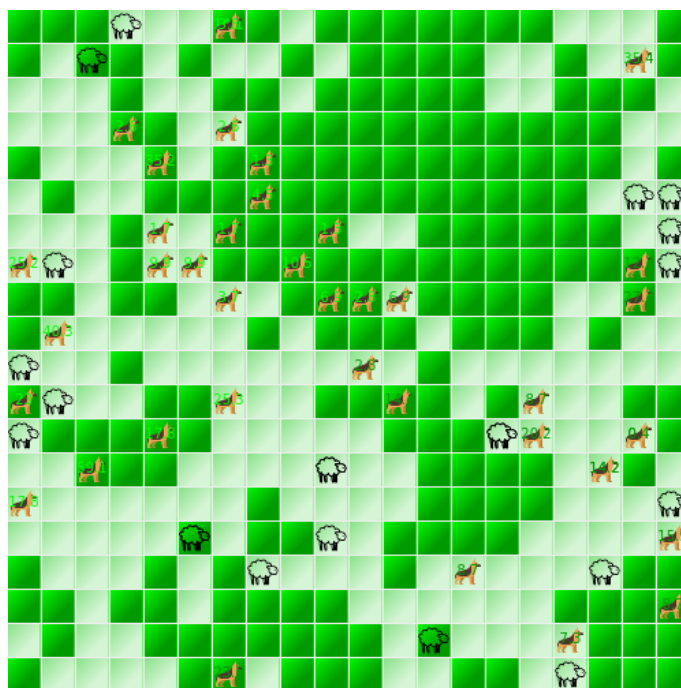
Rys. 2.1: Wykres wielkości populacji w modelu Lotka-Volterra. Źródło [9]

nia mogą mieć bardzo istotny wpływ dla całej populacji. Można rozważać skomplikowane systemy w postaci sieci społecznej, gdzie każdy agent reprezentuje człowieka czy w postaci przestrzeni po której agenci się swobodnie poruszają. Rozważani agenci nie muszą być agentami wirtualnymi i jest możliwość korzystania z fizycznych modeli czy makiet, jednak takie podejście może się okazać kosztowne i nie oferować wystarczającej kontroli czy bezpieczeństwa. Możliwymi zastosowaniami może być rozważenie agentowej wersji wspomnianego modelu Lotki-Volterra (rys. 2.2), symulacja zachowań tłumu w trakcie sytuacji wyjątkowych czy zachowania ekonomiczne.

Innym przykładowym modelem tego typu jest model zachowania stada ptaków w trakcie lotu przedstawiony w pracy Craiga W. Reynolds[23]. W modelu tym indywidualne proste zachowania agentów prowadzą do powstawania skomplikowanych zachowań całego stada. Pojedyncze osobniki zachowują się zgodnie z następującymi zasadami:

- unikanie kolizji - agent unika kolizji z innymi agentami,
- zachowanie prędkości - agent dostosowuje swoją prędkość do swoich sąsiadów,
- centrowanie do grupy - agent stara się poruszać w kierunku centroidy swojego sąsiedztwa.

Powyższy zestaw trzech prostych reguł powoduje powstawanie różnorodnych i skomplikowanych form oraz zachowań stada. Ponadto istnieją rozszerzenia o dodatkowe reguły, które

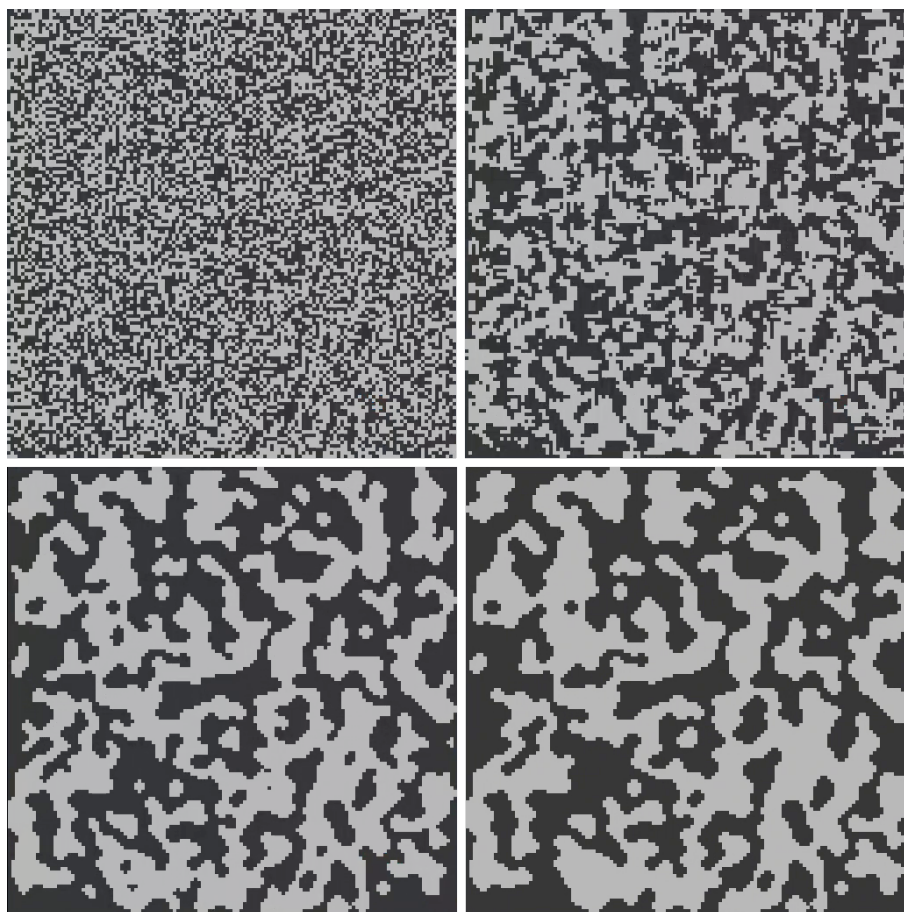


Rys. 2.2: Model Lotki-Volterra w wersji agentowej.

ponownie dotyczą pojedynczych agentów, a mają wpływ na zachowania całości. Taki model często jest wykorzystywany w celu tworzenia efektów wizualnych na potrzeby filmów oraz gier.

Automaty komórkowe pozwalają na symulowanie z wykorzystaniem dyskretnej przestrzeni komórek, gdzie każda komórka znajduje się w określonym stanie i posiada ustalone własności, a ich zmiana odbywa się na podstawie cech sąsiednich komórek. Różne reguły tranzycji pozwalają na generowanie skomplikowanych wzorów oraz struktur. Komórki mogą posiadać wiele klas, przykładowo o cyklicznej zmianie. Ciekawym rozwiązaniem są również rozszerzenia do większej liczby wymiarów. Najczęściej stosowany rodzaj przestrzeni to regularna siatka kwadratów, jednak można też wykorzystać komórki trójkątne lub sześciokątne. Najbardziej znanym przykładem automatu komórkowego jest „Gra w życie” (Conway’s Game of Life), gdzie zmiany stanów komórek są opisane prostymi zasadami opartymi o stan rozważanej komórki oraz licznosci jej sąsiedztwa. W ten sposób możemy rozważać automaty komórkowe jako szczególny przypadek modelu agentowego, gdzie następuje ograniczenie do jednego rodzaju przestrzeni w jakiej znajdują się agenci. Bardzo istotną różnicą pomiędzy modelem agentowym, a opartym o automat komórkowy jest swoboda z jaką można opisywać zasady i działania agentów. W modelu agentowym każdy agent może posiadać osobny i całkowicie różny zestaw reguł i możliwych do podjęcia akcji, gdzie w automacie globalnie dla każdej komórki stosuje się identyczne zasady rozwoju. Z tego powodu w modelu tego rodzaju rozważanie wielu heterogenicznych populacji jest bardzo trudne, a nawet niemożliwe. Dodatkowo nie jest rozważany w tym przypadku ruch agentów w dostępnym środowisku, a jedynie zmiany ich stanów. Przykładowy proces

rozwoju automatu komórkowego został przedstawiony na rysunku 2.3. Początkowe losowe zainicjowanie komórek jednym z dwóch możliwych stanów doprowadziło do powstania stabilnego i przypominającego jaskinie układu.



Rys. 2.3: Rozwój prostego automatu komórkowego. (Opracowanie własne)

## 2.2. MODELE EPIDEMIOLOGICZNE

Obecnie najczęściej używanym i podstawowym modelem epidemii jest model SIR, który opisuje zależności pomiędzy trzema populacjami za pomocą równań różniczkowych. Model ten został opracowany przez Williama Kermacka i Andersona McKendricka, a przedstawiony w pracy „A Contribution to the Mathematical Theory of Epidemics”[15]. W tym modelu istniejące trzy populacje. Równania opisujące ten model to:

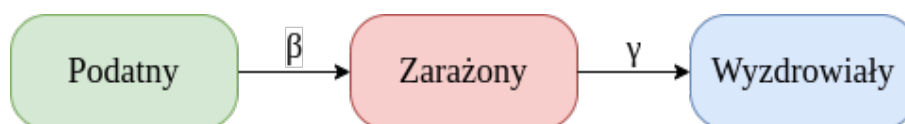
$$\begin{aligned}
 \frac{dS}{dt} &= -\frac{\beta SI}{N}, \\
 \frac{dI}{dt} &= \frac{\beta SI}{N} - \gamma I, \\
 \frac{dR}{dt} &= \gamma I, \\
 N &= S + I + R,
 \end{aligned}
 \tag{2.1}$$

gdzie poszczególne symbole wyjaśnić można w następujący sposób:

- Susceptible (S) - część populacji podatna na zarażenie. Po polsku nazwę można przetłumaczyć jako osoby podatne,
- Infectious (I) - część populacji już zarażona. Osobniki ze stanu S przechodzą do tego stanu z prawdopodobieństwem  $\beta$ . Są to osoby zarażone,
- Recovered (R) - część populacji, która nie podlega już możliwości zarażenia i przebyła chorobę. W ramach tej kategorii uwzględnia się osoby, które nabyły odporność oraz osoby zmarłe, które z oczywistych względów już się nie zarażą. Przejście do tego stanu jest zależne od współczynnika  $\gamma$ , a jego odwrotność oznacza średnią liczbę dni potrzebnych do wyzdrowienia. Po polsku jest to populacja wyzdrowiała,
- N - całkowita liczebność populacji. Jest to wartość stała dla tego modelu.

Stany i przejścia pomiędzy kolejnymi stanami zostały przedstawione na rysunku 2.4. W podstawowej wersji rozważa się stałą populację, dlatego jednym z możliwych rozwinięć jest dodanie zmienności populacji. Kolejnym możliwym rozszerzeniem może być uwzględnienie utraty odporności. Parametrem jaki również można rozważać jest bazowy współczynnik reprodukcji  $R_0$ . Oznacza on średnią liczbę nowych osób, które zakazi osoba zarażona. Wartości mniejsze niż 1 oznaczają wygasanie epidemii, a większe niż 1 rozwijanie się jej. Wzór za pomocą którego można go wyrazić to:

$$R_0 = \frac{\beta}{\gamma}. \quad (2.2)$$



Rys. 2.4: Stany w modelu SIR. Opracowano na podstawie [3].

Trochę bardziej złożonym podejściem jest wykorzystanie modelu SEIR. W ramach niniejszej pracy postanowiono wykorzystać ten model ze względu na możliwość bardziej realistycznego oddanie rozwoju rzeczywistej epidemii COVID-19. Model ten posiada dodatkowy stan, który oznacza iż dany osobnik jest już zarażony ale jeszcze niezdolny do

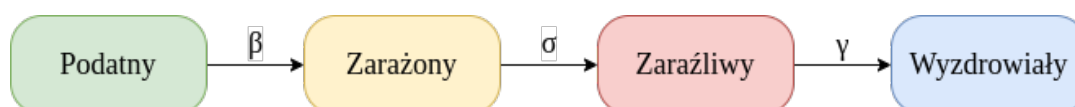
zarażania innych. Model ten można opisać następującym zestawem równań:

$$\begin{aligned}
 \frac{dS}{dt} &= -\frac{\beta SI}{N}, \\
 \frac{dE}{dt} &= \frac{\beta SI}{N} - \sigma E, \\
 \frac{dI}{dt} &= \sigma E - \gamma I, \\
 \frac{dR}{dt} &= \gamma I, \\
 N &= S + E + I + R,
 \end{aligned} \tag{2.3}$$

gdzie większość oznaczeń jest identyczna jak dla modelu SIR, jednak należy wyróżnić następujące różnice:

- Exposed (E) - część populacji już wystawiona na działanie wirusa, ale nie zdolna jeszcze do dalszego jego rozprzestrzeniania. Przykładowo może się to wiązać z występowaniem objawów dopiero po jakimś czasie. Osobniki ze stanu S przechodzą do tego stanu z prawdopodobieństwem  $\beta$  tak jak w modelu SIR,
- parametr  $\sigma$  - współczynnik przejścia ze stany Exposed do stanu Infectious, którego odwrotność oznacza średnią liczbę dni potrzebną do przejścia do stanu zaraźliwego.

Znaczenia pozostałych parametrów oraz stanów nie ulegają zmianie. Model ten pozwala na wzięcie pod uwagę osobników ukrytych, u których już rozwija się choroba. Graficznie stany i przejścia przedstawiono na rysunku 2.5. W tym przypadku również można stosować wcześniej wspomniane rozszerzenia.



Rys. 2.5: Stany w modelu SEIR. Opracowano na podstawie [2].

### 2.3. PAKIETY DO MODELOWANIA AGENTOWEGO

Obecnie bardzo wiele zastosowań mają obliczenia wykonywane na komputerach, a ich możliwości do przeprowadzania obliczeń znacznie wyprzedzają zdolności ludzi, dlatego można je wykorzystać w celu modelowania agentowego. Wśród przykładowych rozwiązań informatycznych można wyróżnić:

- NetLogo <sup>1</sup>,
- MASON <sup>2</sup>,

<sup>1</sup> <https://ccl.northwestern.edu/netlogo/>

<sup>2</sup> <https://cs.gmu.edu/~eclab/projects/mason/>

- Mesa<sup>3</sup>,
- Agents.jl<sup>4</sup>.

NetLogo jest projektem rozwijanym przez Uri Wilensky na Northwestern University. Środowisko jest wydane na licencji GPL i napisane w językach Scala oraz Java, a tym samym działa na JVM<sup>5</sup>. Możliwe jest wykorzystywanie modeli dwuwymiarowych oraz trójwymiarowych. Dodatkowo środowisko posiada bardzo bogatą bibliotekę przykładowych modeli. MASON jest rozwijany na George Mason University. MASON jest napisany w języku Java. Z pakietem dostępne jest wiele przykładowych modeli, zarówno dwuwymiarowych, jak i trójwymiarowych. Mesa jest pakietem do modelowania agentowego dla języka Python. Dzięki takiemu podejściu możliwe jest wykorzystanie również szerokiego wachlarza innych pythonowych bibliotek do analizy danych, uczenia maszynowego czy szeroko pojętej danologii. Znaczną zaletą tej biblioteki jest możliwość w pełni swobodnego definiowania zachowań oraz stanów agentów, dzięki czemu można w prosty sposób osiągnąć wymagane efekty. Agents.jl jest biblioteką do modelowania agentowego dla języka programowania Julia. Biblioteka ta nie jest zbyt popularna, jednak posiada wiele funkcjonalności, bogatą dokumentację oraz dostarcza wiele przykładów. Jedną z ciekawszych właściwości jest integracja z Open Street Map, dzięki temu można wykorzystywać istniejące prawdziwe mapy jako środowisko dla agentów. Autorzy wskazują też na znaczną szybkość działania swojej biblioteki, która według ich zapewnień jest około 30 razy wydajniejsza niż Mesa, co pozwala na przeprowadzanie bardziej rozbudowanych symulacji na identycznej konfiguracji sprzętowej. Porównanie wymienionych środowisk przedstawiono w tabeli 2.1. Z wszystkich wymienionych rozwiązań jedynie Mesa i Agents.jl pozwalają na łatwą integrację modelu z bibliotekami służącymi do optymalizacji i analizy danych, jednak język Python jest prostszy w użyciu i posiada większą społeczność, dzięki czemu uzyskanie pomocy jest prostsze. Na podstawie analiz postanowiono w ramach niniejszych pracy wykorzystać pakiet Mesa i z tego powodu został on omówiony dokładniej w rozdziale 2.4.

---

<sup>3</sup> <https://mesa.readthedocs.io/en/master/>

<sup>4</sup> <https://juliadynamics.github.io/Agents.jl/stable/>

<sup>5</sup> JVM - Java Virtual Machine

Właściwość	NetLogo	MASON	Mesa	Agents.jl
Typy parametrów	ograniczone do kilku typów	dowolne	dowolne	dowolne
Analiza i modelowanie w tym samym języku	nie	Java	Python	Julia
Integracja z innymi bibliotekami	trudne w użyciu	trudne w użyciu	dowolny	dowolny
kontrola poprzez GUI	tak	zależne od użytkownika	zależne od użytkownika	brak

Tabela 2.1: Porównanie niektórych właściwości wymienianych pakietów służących do symulacji agentowej.

## 2.4. OPIS ŚRODOWISKA MESA

W celu przeprowadzenia symulacji agentowej wykorzystano środowisko Mesa[14]. Rozwój projektu został zapoczątkowany w 2013 roku i w grudniu 2014 w wersji 0.5 trafił do repozytorium paczek PyPI<sup>6</sup>. Główne komponenty pakietu to modelowanie, analiza i wizualizacja symulacji. Modelowanie w środowisku Mesa przebiega niezwykle łatwo z wykorzystania języka programowania Python. Takie połączenie zapewnia szybkość implementacji. Python jest również jednym z wiodących języków służących do analizy danych i uczenia maszynowego, a dzięki jego wykorzystaniu bezproblemowo można wykorzystywać Mesę razem z innymi popularnymi pakietami. Mesa pozwala na w pełni swobodne definiowanie akcji oraz stanów agentów z wykorzystaniem języka Python.

W celu przygotowania symulacji należy stworzyć klasy odpowiadające agentom oraz wykorzystywanemu modelowi. Obie klasy należy rozszerzyć z klas dostarczanych przez środowisko Mesa. Można wyróżnić następujące metody aktywacji agentów:

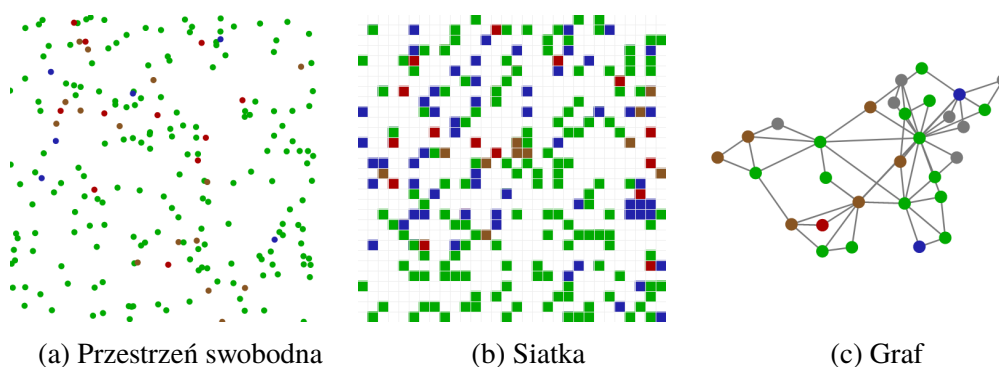
- aktywacja podstawowa - agenci wykonują swoje akcje w kolejności w jakiej zostali dodani do symulacji,
- aktywacja losowa - wybierany jest losowy agent, który nie wykonał jeszcze akcji w danym kroku symulacji,
- aktywacja jednoczesna - dwuetapowa metoda aktywacji, gdzie najpierw wszyscy agenci wykonują akcję opisaną przez pierwszy etap, a dopiero następnie wykonywana jest pozostała część działania agenta. Przykładowo w pierwszym etapie można dokonać ruchu agentów, a w drugim aktualizacji ich stanów,
- aktywacja wieloetapowa - po kolei wykonywana jest lista etapów, a w ramach każdego z nich wykonywana jest akcja danego agenta powiązana z danym etapem.

<sup>6</sup> PyPI - Python Package Index

Agenci muszą podejmować działania w jakimś środowisku i w tym celu Mesa udostępnia szereg wbudowanych przestrzeni. Można wyróżnić modele środowisk dyskretnych i ciągłych. Dodatkowo przestrzenie mogą opcjonalnie się zawiązać na krawędziach. Dostępne typy przestrzeni symulacji w Mesa:

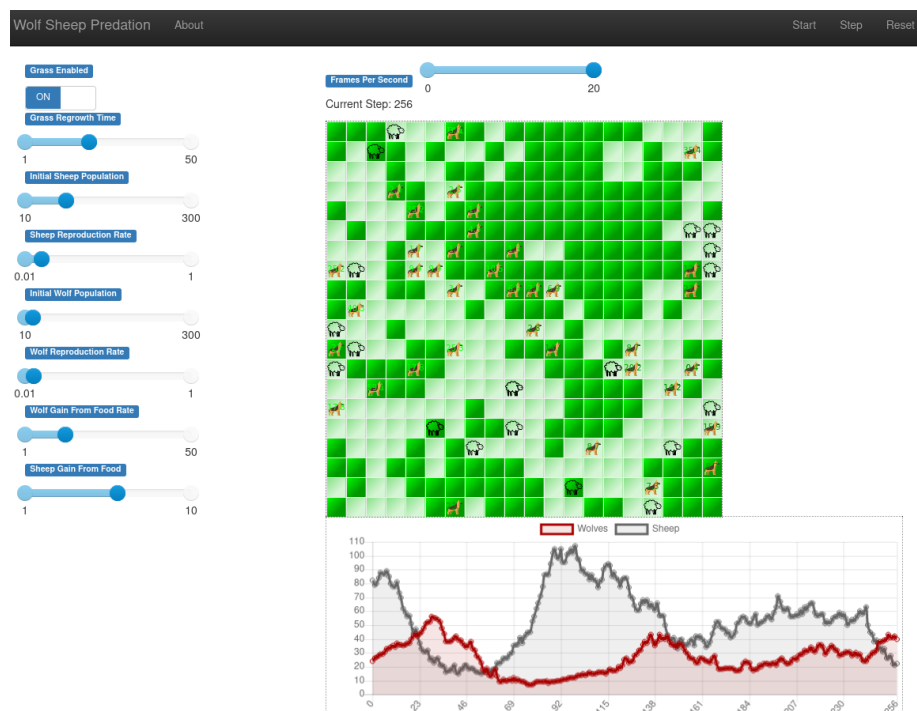
- przestrzeń ciągła - w tym przypadku agenci poruszają się swobodnie z ciągłymi współrzędnymi, a interakcje z innymi agentami odbywają się w zakresie zdefiniowanej odległości. Przykładowy wygląd został przedstawiony na rysunku 2.6a,
- regularna siatka - przestrzeń z dyskretnymi wartościami współrzędnych, gdzie odległość definiowana jest w ilości komórek. Można ją wykorzystać w celu wytworzenia automatu komórkowego. Przykład zaprezentowano na rysunku 2.6b. Mesa dostarcza trzy specyficzne rodzaje tej przestrzeni:
  - siatka pojedyncza - siatka, gdzie w danej komórce może się znajdować co najwyżej jeden agent,
  - siatka wielokrotna - siatka, gdzie w danej komórce może się znajdować dowolna liczba agentów,
  - siatka heksagonalna - siatka, gdzie komórka ma kształt sześciokąta, a tym samym ma sześciu sąsiadów,
- graf - komórki stanowią węzły grafu, gdzie w każdej może się znajdować zero lub więcej agentów. Ten rodzaj przestrzeni został przedstawiony na rysunku 2.6c.

Mesa w celu wizualizacji symulacji wykorzystuje środowisko przeglądarki z frameworkiem Tornado po stronie backendu. Istnieje możliwość renderowania dostarczanych przez Mesę przestrzeni oraz agentów z możliwością użycia kolorów i obrazów. Ponadto można pisać własne komponenty wizualizacji przy wykorzystaniu HTML oraz języka JavaScript. W webowym widoku symulacji znajdują się też kontrolki, do sterowania parametrami symulacji. Ponadto istnieje możliwość rysowania wykresów dla danych zebranych w trakcie symulacji. Wymienione funkcjonalności można zauważyć na rysunku 2.7.



Rys. 2.6: Przykłady dostępnych przestrzeni.





Rys. 2.7: Przykładowy widok okna Mesa. Można wyróżnić: widok symulacji, generowane wykresy oraz kontrolki.

### 3. ANALIZA PROPONOWANEGO ROZWIĄZANIA

W niniejszym rozdziale przedstawiono szereg analiz istniejącej literatury związanej z COVID-19. Zaczęto od przedstawienia analiz danych medycznych w celu przedstawienia najbardziej prawdopodobnych wartości estymowanych parametrów. Zebrane w ten sposób informacje posłużą do sprawdzenia poprawności proponowanego rozwiązania. Wskazano i krótko opisano źródła danych dotyczących przypadków zarażeń i rozwoju epidemii w poszczególnych krajach. Przedstawiono istniejące rozwiązania i prace związane z modelowaniem epidemii COVID-19. Część z nich skupia się na estymacji parametrów, a część na modelowaniu agentowym. Dokładnie opisano rozwiązanie proponowane w ramach niniejszej pracy i porównano je z przedstawionymi pracami. Zaproponowany został również plan badań. Podzielony on został na badania wstępne, które mają za zadanie sprawdzenie wpływu różnych czynników i podejść na wartość funkcji błędu oraz na badania ewaluacyjne. Badania ewaluacyjne na podstawie wcześniejszych wyników mają za zadanie przeprowadzić ostateczną estymację parametrów, które zostaną porównane z wartościami pochodzącymi z literatury.

#### 3.1. PARAMETRY COVID-19

W problemie estymacji parametrów epidemii COVID-19 należy rozpatrzyć istniejące już ustalenia wynikające z zebranych i przetworzonych danych medycznych dotyczących poszczególnych pacjentów. Wartości zebrane w ramach tego rozdziału posłużą w celu weryfikacji poprawności wyników końcowych. Dla rozpatrywanego modelu epidemii SEIR są to 3 parametry -  $\beta$ ,  $\sigma$ ,  $\gamma$ , których znaczenie zostało wyjaśnione w rozdziale 2.2. Niektórzy autorzy mogą podawać parametry jako odpowiednie ilości dni, które są odwrotnością parametrów  $\sigma$  oraz  $\gamma$ . Dodatkowo należy zebrać informacje o śmiertelności. Istotnym czynnikiem wpływającym w dużym stopniu na sposób zakończenia przebiegu choroby jest wiek pacjenta, dlatego dane o śmiertelności należy uwzględnić z podziałem na kategorie wiekowe. Sposoby podziału mogą się różnić w zależności od autora. Kolejnymi istotnymi czynnikami są testy, wykrywalność choroby oraz liczność pominiętych przypadków. W oficjalnych danych podawane są tylko przypadki potwierdzone pozytywnym testem, jednak zakres testowanych osób zależy od podejścia danego państwa w danym czasie. Nie można mówić o jednakowym dostępie do testów na obecność SARS-CoV-2 w skali globalnej. Testy nie obejmują całej populacji, a tylko część która ze względu na przepisy musi posiadać wyniki testu albo miała objawy kwalifikujące do wykonania go. Należy uwzględnić również

przypadki bezobjawowe, których wykrycie jest prawie niemożliwe. Problem stanowią ponadto osoby, które pomimo możliwości bycia zarażonym SARS-CoV-2 i wykazujących charakterystyczne objawy, nie badają się ze względu na strach oraz niechęć do potencjalnej kwarantanny, która byłaby dla nich uciążliwa. W celu znalezienia odpowiednich danych statystycznych dotyczących wymienionych zagadnień dokonano przeglądu literatury.

W pracy [11] z kwietnia 2020 roku przedstawiono dane statystyczne dotyczące 1099 pacjentów pochodzących z 30 różnych rejonów Chin zebranych do 29 stycznia 2020r. Artykuł ten stanowi bogate źródło informacji o COVID-19 już od wczesnego etapu pandemii, gdyż wcześniejsza jego wersja ukazała się już w lutym. Średni czas inkubacji wirusa autorzy określili na 4 dni z rozstępem międzykwartylowym od 2 do 7. W pracy [16] z 26 marca przeanalizowano dane dotyczące 425 pacjentów z Wuhan. Autorzy określili średni czas inkubacji wirusa na 5,2 dnia z 95% przedziałem ufności w zakresie 4,1 do 7 dni. 95. percentyl wypadł na 12,5 dnia, a parametr reprodukcji został określony na 2,2. W pracy [1] przeanalizowano dane dotyczące jednych z pierwszych przypadków z Wuhan. Autorzy czas inkubacji określili z wykorzystaniem rozkładu Weibulla, gamma oraz logarytmicznonormalnego, gdzie rozkład Weibulla najlepiej dopasowywał się do danych ze średnią 6,4. W pracy [18] porównano wyniki przedstawione przez wielu autorów. Czas trwania inkubacji zamodelowano przy pomocy rozkładu logarytmicznonormalnego ze średnią 5,8 dnia. Inne wyniki otrzymano w pracy [24]. Średni czas do śmierci określili na 17,8 dnia, a czas potrzebny na wyjście ze szpitala na 24,7 dni. Ze względu na to, że model SEIR nie rozróżnia sposobu zakończenia choroby należy przyjąć wartość parametru  $\gamma$  równą 17,8. Europejskie Centrum ds. Zapobiegania i Kontroli Chorób podaje, że czas od kontaktu z wirusem do objawów wynosi od 1 do 14 dni, a czas potrzebny na wyzdrowienie to 8 do 10 dni, jednak w przypadkach ciężkich ten czas wynosi do dwóch tygodni[7]. Centers for Disease Control and Prevention (CDC) podaje, że okres inkubacji wirusa trwa do 14 dni, a zazwyczaj jest to 4 lub 5 dni[5]. Parametry związane z modelem SEIR oraz ich przedziały ufności lub innego rodzaju zakresy w zależności od autorów zebrano w tabeli 3.1 z podziałem na źródła. W żadnej z wymienionych publikacji nie był podawany parametr  $\beta$ , który zależy mocno od działań prewencyjnych i zachowań ludzi, a z medycznego punktu widzenia nie jest tak istotny jak pozostałe. Przedstawione wartości posłużą w celu walidacji wyników otrzymanych w ramach niniejszej pracy.

Źródło	$\beta$	PU $\beta$	$\sigma$	PU $\sigma$	$\gamma$	PU $\gamma$	$R_0$
Guan[11]	-	-	4	2-7 <sup>a</sup>	-	-	-
Li[16]	-	-	5,2	4,1-7	12,5	10,3-14,8	2,2
Backer[1]	-	-	6,4	5,6-7,7 <sup>b</sup>	-	-	-
McAloon[18]	-	-	5,8	5,0-6,7 <sup>b</sup>	-	-	-
Verity[24]	-	-	-	-	17,8	16,9-19,2 <sup>b</sup>	-
ECDC[7]	-	-	-	1-14 <sup>c</sup>	-	8-14 <sup>c</sup>	-
CDC[5]	-	-	5	4-14 <sup>c</sup>	-	-	-

a - autorzy podali rozstęp międzykwartylowy

b - autorzy podali przedział wiarygodności

c - autorzy podali przedział możliwych wartości

Tabela 3.1: Zebrane parametry dla modelu epidemii SEIR z ich przedziałami ufności dla różnych źródeł.

Ważnym aspektem choroby jest śmiertelność. Istotne jest jej określenie w celu ustalenia konkretnych strategii kontrolnych oraz ochronnych społeczeństwa, a w szczególności osób najbardziej narażonych na tak przykry wynik zachorowania. Jednym z czynników wpływającym na śmiertelność jest wiek danej osoby. W celu ustalenia współczynników śmiertelności dla poszczególnych grup wiekowych dokonano przeglądu dostępnych źródeł informacji. Na podstawie dostępności danych i istotności śmiertelności został również dobrany podział na grupy wiekowe. W pracy [24] przedstawiono analizę przypadków w chińskiej prowincji Hubei z uwzględnieniem wpływu cenzury oraz niepewności stwierdzania zarażeń. Wyniki przedstawione w tej publikacji podsumowano w tabeli 3.2. Uwzględniono również podział na osoby młodsze i starsze niż 60 lat, gdzie śmiertelność wynosi odpowiednio 0,318% i 6,38%. Autorzy pracy [29] przeanalizowali raport opublikowany przez Chińskie Centrum Kontroli i Zapobiegania Chorobom i na jego podstawie przedstawili między innymi śmiertelność. Dla osób w wieku do 9 lat nie było żadnego przypadku śmiertelnego, w grupie 70-79 śmiertelność wyniosła 8%, a dla osób mających więcej niż 80 lat aż 14,8%. Innym źródłem informacji o śmiertelności w poszczególnych grupach wiekowych może być amerykańskie CDC[5], które podaje wartości dla USA. Wartości pochodzące z tej instytucji zebrano i opracowano w tabeli 3.3. W celu uwzględnienia wieku zarażonych w badaniach symulacyjnych postanowiono przyjąć następujące wartości śmiertelności opracowane na podstawie przedstawionej literatury.

- <60 lat - 0,32%,
- 60-69 - 3,5%,
- 70-79 - 8,61%,
- $\geq 80$  - 14%.

Wiek	Śmiertelność
0-9	0,0026%
10-19	0,0148%
20-29	0,06%
30-39	0,146%
40-49	0,295%
50-59	1,25%
60-69	3,99%
70-79	8,61%
≥80	13,4%
<60	0,318%
≥60	6,38%

Tabela 3.2: Śmiertelność w poszczególnych grupach wiekowych wg [24].

Wiek	Śmiertelność - zakres	Śmiertelność - średnia
≥85	10-27%	18,5%
65-84	3-11%	7%
55-64	1-3%	2%
<55	<1%	<1%

Tabela 3.3: Śmiertelność w poszczególnych grupach wiekowych wg [5].

Istotną rzeczą jest również sama wykrywalność choroby. Zależy ona od dostępu do testów oraz od udziału osób bezobjawowych w całości populacji chorych. W celu określenia jak dużą część stanowią przypadki ukryte przeprowadzono analizę literatury. Na podstawie analizy zachorowań na statku Diamond Princess określono licznosc grupy asymptotycznej na 17,9% z przedziałem wiarygodności 15,5–20,2% na poziomie ufności 95%[20]. Są to jednak wczesne wyniki określone na ściśle kontrolowanej grupie pasażerów wycieczkowca i z tego powodu należy poszukać również innych źródeł. Inni autorzy określili proporcje całkowitej liczby przypadków do liczby wykrytych zachorowań dla sześciu krajów europejskich[4]. Dla każdego z nich wartości oscylowały wokół 2,3 i taka wartość została przyjęta za proporcję wszystkich przypadków do tych wykrytych. Oznacza to, że na każdego wykrytego chorego przypadają 1,3 przypadki niewykryte. Taka proporcja przekłada się na to, że około 56% przypadków nie zostaje wykryta. Warto zaznaczyć, że jest to dolne ograniczenie na tę wartość i może być ona większa. Analizy te przeprowadzono na danych z marca oraz kwietnia. W celu poznania liczby przypadków ukrytych można skorzystać z analizy bayesowskiej, jak to zrobiono dla zarażeń w USA[28]. Autorzy określili, że na dzień 18 kwietnia 2020 liczba rzeczywistych przypadków była 9 razy większa niż wynika

to z danych oficjalnych, a biorąc pod uwagę niepewność wychodzi, że jest ich od 3 do nawet 20 razy więcej.

### **3.2. ŹRÓDŁA DANYCH**

W internecie można znaleźć wiele źródeł danych o COVID-19. Są to zarówno źródła krajowe jak i światowe. Wiele instytucji oraz projektów zajmuje się agregowaniem danych. W ramach niniejszej pracy postanowiono wykorzystać dane udostępnione przez Our World in Data[17]. Dane są agregowane z wielu różnych źródeł. Wśród udostępnionych danych znajduje się aż 59 różnych informacji. Przykładowo można wyróżnić: kod ISO kraju, liczbę nowych i całkowitych przypadków, śmierci, testów czy szczepień. Dane są dostępne w formacie csv, co pozwala na bezproblemowe wykorzystywanie ich z oprogramowania służącym do analizy danych. Niestety wśród dostępnych informacji nie znajdują się dane przedstawiające dzienną liczbę wyzdrowiałych osób, a te dane mogą również posłużyć w celu obliczania błędu modelu względem danych rzeczywistych. Z tego powodu dane uzupełniono informacjami pochodzącymi od Center for Systems Science and Engineering na uniwersytecie Johns Hopkins, które dostępne są w repozytorium w serwisie Github[6]. Wśród dostępnych danych znajdują się dane surowe z kolejnych dni oraz przygotowane pliki csv tworzące szeregi czasowe. Ze względu na łatwiejsze wykorzystanie oraz wstępne przygotowanie zdecydowano się na wykorzystanie gotowych szeregów czasowych. W ramach pracy postanowiono również wykorzystać strukturę wieku społeczeństw. Dane dotyczące liczności ludności w poszczególnych grupach wiekowych dostępne są na stronach internetowych ONZ[22]. Dane przedstawiają między innymi stan na 2019 rok i nie powinno być znacznej różnicy względem roku 2020 lub 2021. Grupy wiekowe są podzielone na kategorie co 5 lat, dzięki czemu można dokładnie dobrać przedziały w symulacji.

### **3.3. ISTNIEJĄCE ROZWIĄZANIA**

Warto dokonać również analizy literatury w zakresie istniejących rozwiązań, których zadaniem jest estymacja parametrów epidemii COVID-19. Bez problemu można znaleźć wiele prac związanych z tym tematem, ze względu na globalny zasięg pandemii oraz jej znaczny wpływ na funkcjonowanie ludzkości w wielu aspektach. Różni autorzy mogą podchodzić do tego zagadnienia na wiele zróżnicowanych sposobów. Przykładowo mogą brać pod uwagę model typu SIR albo SEIR czy też dokonywać znacznych rozszerzeń w celu uwzględnienia kwarantanny, hospitalizacji. Możliwe jest też operowanie na modelu matematycznym ale też agentowym.

W pracy [12] zaprezentowano wykorzystanie metody roju cząstek w celu estymacji parametrów modelu SEIR na danych pochodzących z prowincji Hubei. Autorzy rozszerzyli model o dodatkowe stany, takie jak: kwarantanna, hospitalizacja, wykryci oraz niewykryci zarażeni. Zostały wykorzystane dane pochodzące z oficjalnej strony komisji zdrowia Wuhan.

Metoda roju cząstek została wykorzystana do otrzymania parametrów opisujących częstość konieczności hospitalizacji oraz współczynnik wychodzenia z kwarantanny. Pozostałe parametry pochodzą z literatury i są stałe.

W pracy [10] przedstawiono predykcję rozwoju epidemii COVID-19 we Włoszech. W tym celu dokonano estymacji parametrów matematycznego modelu SEIR z wykorzystaniem metody roju cząstek oraz dopasowania krzywej przy pomocy metody najmniejszych kwadratów, a następnie porównano wyniki otrzymywane przy pomocy obu podejść. Wykorzystano dane gromadzone przez uniwersytet Johna Hopkinsa oraz dane dotyczące mobilności za pośrednictwem Google. W pracy dokonano rozszerzenia klasycznego modelu SEIR o następujące stany: kwarantanna pomiędzy stanem zaraźliwym oraz następnymi, śmierć oraz stan całkowitej niewrażliwości na chorobę. Taka konstrukcja modelu daje łącznie 6 różnych parametrów, które podlegają estymacji. W celu oceny jakości modeli wykorzystano znormalizowany błąd średniokwadratowy obliczany na podstawie dopasowania ilości przypadków potwierdzonych, wyzdrowiałych i śmiertelnych. Dla metody roju cząstek jej działanie powtórzono 50 razy ze względu na losowe inicjowanie oraz działanie metody. Wartości błędów otrzymywane za pomocą metody roju cząstek były nieznacznie mniejsze niż dla metody najmniejszych kwadratów.

Analizy rozwoju epidemii COVID-19 można dokonywać nie tylko za pomocą metod opartych o klasyczne modele epidemiologiczne, ale też za pomocą sztucznych sieci neuronowych, co zostało przedstawione w pracy [30]. Autorzy w swoich badaniach wykorzystali dane publikowane przez Narodową Komisję Zdrowia Chin, informacje o podróżach oraz dane dotyczące epidemii SARS. Przedstawili model SEIR z dynamicznie zmieniającą się liczebnością populacji, co związane jest z przemieszczaniem się ludności. Estymowane były parametry  $\beta$  i  $\gamma$ , a wartość  $\sigma$  została ustalona na 7 na podstawie analizy literatury. Parametr  $\beta$  został wyestymowany na wartości 0,15747 dla przypadków z symptomami oraz na 0,78735 dla przypadków asymptomatycznych. Z kolei  $\gamma$  przyjął wartość 0,154 (95% PU: 0,0721–0,238), co przekłada się na 6 i pół dnia. Do predykcji zarażeń została również wykorzystana sztuczna rekurencyjna sieć neuronowa typu LSTM<sup>1</sup>, a w trakcie jej uczenia został wykorzystany błąd średniokwadratowy. Predykcje rozwoju epidemii zostały porównane dla obu zaproponowanych modeli.

W pracy [27] przedstawiono analizę i estymację parametrów modelu SEIR dla rozwoju epidemii w Wuhan ze szczególnym uwzględnieniem przewidzenia dnia zakończenia kryzysu. Autorzy wykorzystali dane pochodzące z oficjalnej strony internetowej komitetu zdrowia prowincji Hubei i dokonywali estymacji parametrów  $\sigma$ ,  $\gamma$  oraz  $R_0$ . Optymalne wartości parametrów zostały uzyskane przy pomocy metody przeglądu zupełnego. Wartości parametrów były próbkowane w zadanych ograniczeniach z niewielkim krokiem zmian ich wartości. Błąd dopasowania modelu do danych rzeczywistych był obliczany jako suma wartości funkcji błędów na populacji zarażonej oraz populacji wyzdrowiałej, a samą zastoso-

---

<sup>1</sup> LSTM - Long Short Term Memory

wana funkcją był pierwiastek z średniej kwadratowej błędów. Finalne otrzymane wartości parametrów to  $R_0 = 1,44$ ,  $\sigma = 1/3$ ,  $\gamma = 1/14$ .

Wskazać można też istniejące rozwiązania wykorzystujące podejście agentowe do modelowania rozwoju epidemii COVID-19, jak to zostało wykonane dla Francji[13]. Autorzy wykorzystali skomplikowany model agentowy obejmujący aż 194 parametry w celu symulacji wpływu lockdownu na rozwój epidemii. Większość parametrów została zaczerpnięta z analizy literatury, a parametry opisujące ryzyko zarażenia oraz licznosc osób niezdiagnozowanych zostały dopasowywane z wykorzystaniem metody Nelder–Meada z 30 różnymi punktami startowymi w celu wyjścia z minimum lokalnych. W badaniach wyniki modelu były uśrednianie z 200 uruchomień, a w celu zasymulowania 67 milionowej populacji Francji wykorzystano pół miliona agentów.

W przedstawionej przykładowej literaturze można wyróżnić kilka powtarzających się zadań i wykorzystanie innych podejść do ich rozwiązania. Często podstawowy model SIR albo SEIR są rozszerzane o dodatkowe stany, które wynikają z charakterystyki samej choroby oraz działań podejmowanych przez rządy. Kolejną rzeczą jest podejście do estymacji. Wykorzystywane jest dopasowanie krzywej generowanej przez model do danych rzeczywistych, jednak zastosowane metody optymalizacji, która ma zmniejszyć różnice pomiędzy nimi, są różne. W przedstawionych pracach można wskazać metodę najmniejszych kwadratów, Nelder–Meada, roju cząstek i przeglądę zupełnego. W opisanych pracach podano również sposób obliczania jakości modelu. W tym celu można wykorzystać różne dostępne dane, takie jak licznosc zarażonych i wyzdrowiałych lub dodać do tego jeszcze przypadki śmiertelne. W znacznej większości przeanalizowanej literatury w celu estymacji wykorzystywany był model matematyczny, a model agentowy pojawia się głównie w celach symulacji i przewidzenia końca epidemii czy wpływy wprowadzanych obostrzeń.

### 3.4. OPRACOWANE ROZWIĄZANIE

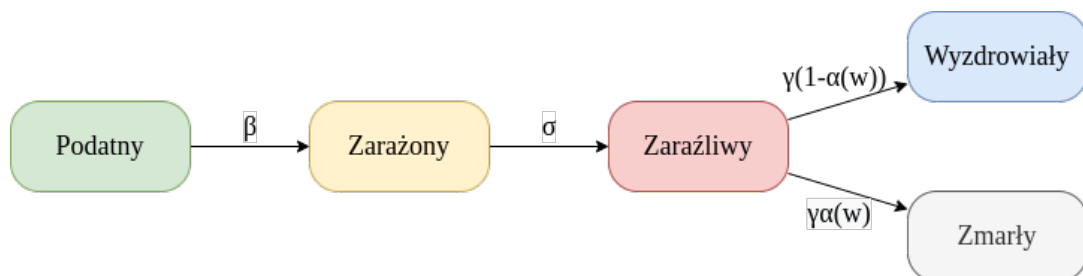
W celu estymacji parametrów epidemii typu SEIR za pomocą modelu agentowego należy przedstawić model stanów i przejść pomiędzy nimi. W celu konstrukcji modelu agentowego wykorzystano klasyczny matematyczny model SEIR i dokonano dekompozycji opisujących go równań w celu zastosowania w podejściu opartym o interakcje indywiduów. W przypadku COVID-19 można zastosować wiele rozszerzeń z uwzględnieniem bardzo wielu szczegółów związanych z podjętymi działaniami mającymi pomóc w walce z epidemią oraz charakterystyka samej choroby. Ważnym czynnikiem jest również śmiertelność, a ta w znaczny sposób zależy od wieku w jakim jest chory. Z tego powodu zdecydowano w modelu uwzględnić nowy stan, który odpowiada za osoby zmarłe. Wartość parametru odpowiadającego za przejście do tego stanu zależy od przedziału wiekowego, do którego należy dany agent. Śmiertelność została ustalona na podstawie analizy literatury przedstawionej w 3.1 i nie podlega estymacji w ramach rozważanej pracy. Udział poszczególnych grup



wiekowych w populacji został ustalony za pomocą dostępnych danych statystycznych[22]. W proponowanych symulacjach agentowych różni agenci mogą zaczynać w innych stanach. Szczególnie ważne by choć jeden agent znajdował się w stanie zaraźliwym lub wystawionym na działanie wirusa, ponieważ w przeciwnym wypadku nie dojdzie do rozwoju choroby. Z tego powodu początkowo w populacji znajduje się kilka osobników zarażonych.

W zaproponowanym modelu agent może się znajdować w danej chwili w jednym z 5 stanów, które graficznie przedstawiono na rysunku 3.1. Znaczenia parametrów są identyczne jak w modelu SEIR przedstawionym w rozdziale 2.2. Widoczny parametr  $w$  odpowiada za wiek danego agenta, a  $\alpha$  jest funkcją, która dla danego wieku zwraca śmiertelność w grupie wiekowej, do której należy. Lista możliwych stanów jest następująca:

- podatny - agent może się zarazić przy kontakcie z agentem zaraźliwym. Znaczna większość zaczyna w tym stanie,
- Zarażony - agent jest już nosicielem wirusa, jednak nie jest zdolny do zarażania innych,
- zaraźliwy - agent może zarażać innych. Agenci przy zmianie do tego stanu są zliczani w celach wygenerowania dziennej liczby nowych zarażeń,
- wyzdrowiały - agent przebył chorobę i jest już odporny. Agenci przy zmianie do tego stanu są zliczani w celach wygenerowania dziennej liczby wyzdowień,
- zmarły - agent zmarł w wyniku choroby. Agenci przy zmianie do tego stanu są zliczani w celach wygenerowania dziennej liczby śmierci.



Rys. 3.1: Stany i przejścia pomiędzy nimi w proponowanym modelu.

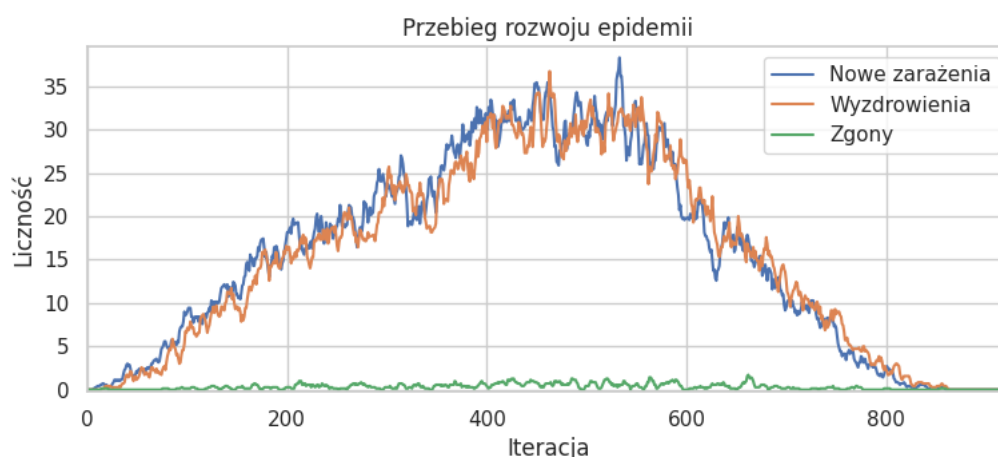
W celu uwzględnienia struktury wieku społeczeństw postanowiono każdemu agentowi przypisać grupę wiekową. Podział na grupy został przygotowany na bazie analizy literatury i postanowiono szczególnie wyróżnić osoby w starszym wieku, gdzie śmiertelność jest znacznie wyższa. Podział na grupy wiekowe wraz z ich śmiertelnościami przedstawia się następująco:

- mniej niż 60 lat - 0,332%,
- pomiędzy 60 i 69 - 3,35%,
- pomiędzy 70 i 79 - 8,61%,
- więcej niż 80 - 14%.

Postanowiono wykorzystać losową aktywację agentów w każdej iteracji. Całość symulacji odbywa się z wykorzystaniem regularnej siatki, gdzie każdą komórkę może zajmować wielu agentów. W każdej iteracji agent może wykonać jedną z kilku akcji:

- pozostać w miejscu,
- przenieść się do sąsiedniej komórki. Nie jest wymagane by była ona wolna,
- jeżeli agent jest w stanie podatnym, to z pewnym prawdopodobieństwem może się zarazić od każdego z zaraźliwych sąsiadów,
- agent z określonym prawdopodobieństwem może spróbować przejść do stanu następnego. Dotyczy to agentów znajdujących się w stanach zarażonym oraz zaraźliwym.

Opracowany model posłuży jako podstawa do estymacji parametrów z wykorzystaniem metod optymalizacji. W trakcie symulacji generowane są miary spotykane przy raportowaniu danych rzeczywistych. Na rysunku 3.2 przedstawiono przykładowy przebieg symulacji i generowanych przez nią wartości.



Rys. 3.2: Przykładowy przebieg symulacji z wykorzystaniem środowiska Mesa.

### 3.5. PLAN BADAŃ SYMULACYJNYCH

Celem badań jest estymacja parametrów modelu epidemii COVID-19. W zagadnieniu estymacji parametrów można rozważyć wiele różnych podejść, jak zostało to wskazane w analizie istniejących rozwiązań w rozdziale 3.3. W ramach pracy oprócz opracowania modelu epidemii COVID-19, postanowiono również zbadać wpływ różnych podejść na otrzymane wyniki estymacji.

Po zapoznaniu się z istniejącymi rozwiązaniami oraz w wyniku analizy możliwych podejść i czynników, które mogą wpłynąć na otrzymane finalne wartości parametrów, opracowany został następujący plan badań:

1. wpływ środowiska wykonawczego,
2. działanie różnych optymalizatorów,

3. wpływ ilości uruchomień na jakość estymacji,
4. zastosowanie różnych funkcji błędu,
5. wpływ transformacji danych,
6. wpływ wektora danych na którym liczony jest błąd,
7. estymacja na różnych zbiorach danych,
8. ewaluacja i weryfikacja otrzymanych wartości.

Wybrane środowisko wykonawcze, jego wersja, czy dla języków kompilowanych użyty kompilator mogą mieć znaczący wpływ na działanie otrzymanego programu, jego zużycie zasobów oraz wydajność. Szybkość wykonania jest kluczowym aspektem w działaniu różnego rodzaju metod optymalizacyjnych czy w zagadnieniach związanych z uczeniem maszynowym, gdzie czas nauki może dochodzić do wielu dni na całym klastrze obliczeniowym. W ramach tego badania uwzględniono następujące środowiska wykonawcze:

— CPython<sup>2</sup>

— PyPy<sup>3</sup>

Wybrano środowisko CPython, które jest standardową implementacją języka Python rozwijaną przez Python Software Foundation oraz społeczność. Środowisko PyPy jest alternatywną implementacją języka Python, której celem jest zwiększenie wydajności programów napisanych w Pythonie. Na stronie projektu twórcy podają, że jest on średnio 4,2 raza szybszy niż CPython dzięki zastosowaniu kompilatora JIT<sup>4</sup>. Taki wzrost wydajności pozwoli na polepszenie parametrów symulacji agentowej oraz samego procesu estymacji z wykorzystaniem optymalizatorów. Autorzy PyPy podają również przypadki w jakich ich środowisko może nie okazać się aż tak wydajne. Są to procesy krótkie, gdzie znaczna część czasu zostanie spożytkowana na samą kompilację, jednak w zastosowaniach tej pracy nie jest to problem. Drugim zagadnieniem jest wykonywanie kodu binarnego, co nie podlega pod kompilację przez PyPy. Dotyczy to biblioteki Numpy, która intensywnie wykorzystuje rozszerzenia napisane w języku C oraz gotowe biblioteki do obliczeń numerycznych takie jak BLAS czy LAPACK. Z wymienionych powodów przewiduje się co najmniej dwukrotny wzrost wydajności przy wykonywaniu kodu w środowisku PyPy względem środowiska CPython. Rozważano również wykorzystanie kompilatora Cython oraz biblioteki Numba. Niestety żadnego z tych dwóch rozwiązań nie udało się wykorzystać do uruchomienia programu. Cython kompiluje kod bez modyfikacji, jednak uruchomienie powoduje błąd dostępu do pamięci. Z kolei Numba nie jest w stanie zaimportować pakietu Mesa. Ponadto w przypadku tych dwóch rozwiązań konieczna byłaby modyfikacja skryptów poprzez dodanie typów lub dekoratorów funkcji dla Numba w celu osiągnięcia lepszych efektów. Z wymienionych powodów postanowiono ograniczyć się do środowisk CPython oraz PyPy.

---

<sup>2</sup> <https://www.python.org/>

<sup>3</sup> <https://www.pypy.org/>

<sup>4</sup> JIT - Just-in-Time

Na otrzymane wyniki oraz czas wykonywania obliczeń znaczący wpływ może mieć zastosowana procedura optymalizująca wartość funkcji błędu. Możliwe podejścia możemy podzielić na algorytmy gradientowe i bezgradientowe, lokalne i globalne. W algorytmach gradientowych wykorzystywane są pochodne optymalizowanej funkcji w celu osiągnięcia optimum, a w rozwiązaniach bezgradientowych wykorzystywana jest wyłącznie informacja o wartościach funkcji. Optymalizatory lokalne mogą szybko osiągać lokalne ekstrema, jednak znalezienie tego globalnego może wymagać uruchamiania z wielu punktów startowych. Inaczej jest w algorytmach globalnych, gdzie próbkowana jest cała dostępna przestrzeń i następuje aktualizacja w kierunku wartości bardziej obiecujących. Wadą takiego rozwiązania jest zazwyczaj dłuższy czas wymagany na potrzeby obliczeń. Często nie mamy pewności osiągnięcia optimum globalnego, ale często zadowalające efekty daje optimum lokalne. Dodatkowo różne metody mogą się różnić szybkością zbieżności do prawidłowego rozwiązania. Z tego powodu warto sprawdzić działanie różnych metod optymalizacyjnych, w celu wytypowania tej najbardziej obiecującej pod względem otrzymywanych wyników oraz czasu działania. Postanowiono sprawdzić działanie następujących algorytmów optymalizujących:

- metoda Powella,
- algorytm L-BFGS-B,
- symulowanie wyżarzanie,
- ewolucja różnicowa,
- rój cząstek (PSO<sup>5</sup>),
- CMA-ES<sup>6</sup>,
- optymalizacja bayesowska,
- przegląd zupełny.

Co do wymienionych metod nie są stawiane żadne założenia dotyczące tego, która metoda uzyska lepsze rezultaty. Na podstawie otrzymanych końcowych wartości parametrów zostanie wybrana jedna z powyższych metod do wykorzystania w dalszych badaniach. Metoda Powell-a oraz BFGS są algorytmami optymalizacyjnymi działającymi lokalnie, dlatego w ich przypadku należy wykonać wiele uruchomień z wieloma różnymi punktami startowymi, co pozwoli na lepsze pokrycie przestrzeni parametrów. Pozostałe wybrane metody działają globalnie, jednak mogą wymagać podania rozwiązania początkowego lub ograniczeń na możliwą przestrzeń poszukiwań. Metoda Powella jest bezgradientową metodą optymalizacji funkcji działającą lokalnie. Implementacja jaką postanowiono wykorzystać w ramach badań dostępna jest w pakiecie SciPy[25]. Postanowiono również sprawdzić algorytm BFGS w wersji z ograniczeniami na możliwe wartości parametrów. Jest to metoda gradientowa. Postanowiono wykorzystać implementację z biblioteki SciPy. Symulowanie wyżarzanie jest

---

<sup>5</sup> PSO - Particle Swarm Optimization

<sup>6</sup> CMA-ES - Covariance matrix adaptation evolution strategy

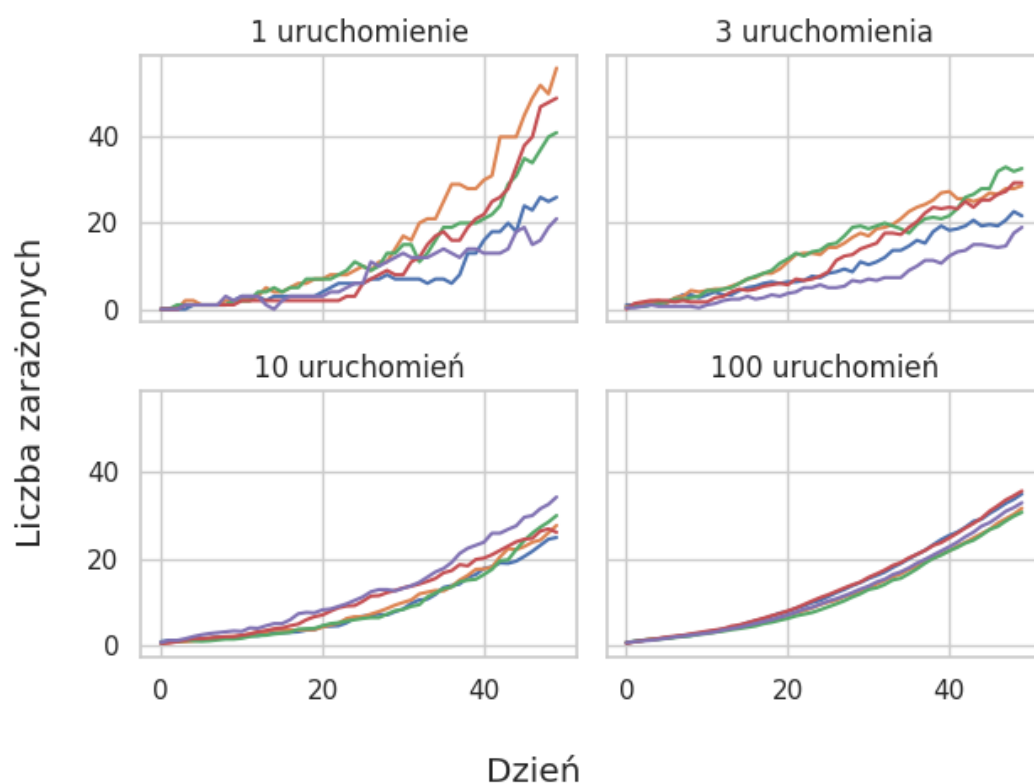
bezgradientową heurystyczną metodą znajdowania globalnego optimum funkcji. Algorytm ten akceptuje rozwiązania gorsze niż obecnie najlepsze, jednak z kolejnymi iteracjami prawdopodobieństwo akceptacji spada. Pozwala to na przejście z fazy eksploracji do fazy eksploatacji rozwiązań najlepszych. Ta metoda również pochodzi z pakietu SciPy pod postacią rozszerzonej wersji nazwanej podwójne wyżarzanie. Ewolucja różnicowa jest bezgradientową metodą optymalizacji globalnej, gdzie przestrzeń możliwych rozwiązań jest eksplorowana przez populację rozwiązań kandydujących, a ich aktualizacja odbywa się za pomocą operatorów znanych z algorytmów genetycznych. W badaniach postanowiono wykorzystać metodę ewolucji różnicowej dostarczaną przez pakiet SciPy. Metoda roju cząstek również jest bezgradientowym, globalnym i opartym o populację algorytmem optymalizacji. Postanowiono wykorzystać istniejącą implementację pochodzącą z biblioteki Pyswarms[19]. Metodę CMA-ES postanowiono wykorzystać z biblioteki DEAP[8]. Polega ona próbkowaniu przestrzeni rozwiązań na podstawie wielowymiarowego rozkładu normalnego, gdzie jego parametry są aktualizowane na podstawie wartości funkcji. Algorytm optymalizacji bayesowskiej optymalizuje globalnie i nie wymaga pochodnych. W trakcie działania statystycznie określa jakie obszary przestrzeni powinny zostać następnie sprawdzone. Postanowiono wykorzystać implementację pochodzącą z biblioteki scikit-optimize[21]. Warto również porównać wymienione metody optymalizacji funkcji do przeglądu zupełnego.

Ze względu na stochastyczną naturę działania modelu agentowego, gdzie działania agentów zależą od losowości, należy uwzględnić wiele wykonanych symulacji. Takie podejście pozwoli zmniejszyć wpływ przypadku na wartość funkcji błędu, a dodatkowo wygładzi ją, co pozwoli na lepsze działanie optymalizatorów. Postanowiono przeprowadzić następujące testy związane z ilością uruchomień symulacji, gdzie ostateczna wartość porównywana do danych rzeczywistych jest średnią z określonej ilości przebiegów:

- uśrednianie z 3 uruchomień,
- uśrednianie z 10 uruchomień,
- uśrednianie z 16 uruchomień,
- uśrednianie z 25 uruchomień.

Można rozważać również większe ilości wykonanych symulacji, jednak oznacza to zwiększenie czasu obliczeń. W idealnym przypadku należałoby ustalić liczbę powtórzeń symulacji na wartość, gdzie dalsze jej zwiększanie nie prowadziłoby do statystycznie istotnego zmniejszania wartości funkcji błędu. Jak widać na rysunku 3.3 zwiększanie liczby powtórzeń powoduje, że generowanie wartości zbiegają się do ustalonego jednego przebiegu. Nie można jednak wziąć wyników symulacji z najmniejszą wartością błędu, ponieważ takie podejście nie oddaje w żaden sposób wyników otrzymywanych w całej populacji możliwych przebiegów symulacji dla danych wartości parametrów. Uśrednianie z wielokrotnych wykonanych powinno pozytywnie wpłynąć na kształt optymalizowanej funkcji błędu poprzez

zmniejszenie wpływu losowości i wprowadzanego przez nią szumu. Optymalizacja powinna przebiegać prościej i bez skakania do wartości już odwiedzonych.



Rys. 3.3: Wpływ ilości uruchomień na powtarzalność wyników. Uśrednianie z wykorzystaniem większej liczby uruchomień zmniejsza rozrzut generowanych przebiegów.

Wybór miary błędu może mieć wpływ na działanie optymalizatora. Różne metody określania błędu mogą mieć ciągle pochodne lub ich nie posiadać. Dodatkową właściwością jaka mogą posiadać miary błędu jest odporność na skalę, gdzie pod uwagę brane są jedynie wartości względne, a tym samym zyskuje się niezależność od wybranej skali, jednostek. Istotny jest również wpływ obserwacji odstających i to jaki mają one wpływa na wartość błędu. Z tego powodu postanowiono sprawdzić działanie optymalizacji z wykorzystaniem następujących miar:

— błąd średniokwadratowy, który przedstawia się w następujący sposób:

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^n (x_i - y_i)^2, \quad (3.1)$$

— średni błąd bezwzględny, wyrażony poniższym wzorem:

$$MAE(x, y) = \frac{1}{N} \sum_{i=1}^n |x_i - y_i|, \quad (3.2)$$

— błąd medianowy, wyrażony następującym wzorem:

$$MedAE(x, y) = \text{mediana}(|x_1 - y_1|, \dots, |x_n - y_n|), \quad (3.3)$$

— logarytmiczny błąd średniokwadratowy, który opisuje się w następujący sposób:

$$MSLE(x, y) = \frac{1}{N} \sum_{i=1}^n (\ln(x_i + 1) - \ln(y_i + 1))^2. \quad (3.4)$$

W podanych wzorach  $N$  oznacza licznosc próby,  $x$  wektor danych generowanych przez model, a  $y$  dane rzeczywiste. Przewiduje się, że zastosowana funkcja nie powinna mieć wpływu na otrzymane wartości parametrów ani na czas obliczeń, ale algorytmy mogą mieć różną tolerancję na zmiany wartości funkcji błędu i na tej podstawie określić czy należy zakończyć działanie.

W celu lepszego i skuteczniejszego przetworzenia danych może być konieczna ich modyfikacja lub przekształcenie na inną formę. W przypadku rzeczywistych danych dotyczących zachorowań istotny wpływ może mieć uśrednienie ich z wykorzystaniem okna przesuwne o pewnym ustalonym rozmiarze. Takie podejście pozwoli usunąć szum występujący w danych, zmniejszy wpływ nagłych skoków wartości i pozwoli lepiej uwzględnić zmiany spowodowane występującą sezonowością, powtarzanymi się charakterystycznymi tygodniowymi zmianami wartości. W tym celu postanowiono wykorzystać uśrednianie z następującymi wielkościami okna:

- 1 - brak uśredniania,
- 3 - niewielka modyfikacja danych,
- 7 - uśrednianie tygodniowe,
- 14 - uśrednianie dwutygodniowe.

Przewidywane jest, że wygładzanie szeregów czasowych powinno mieć pozytywny wpływ na wartość funkcji błędu, a tym samym na estymację parametrów. Jednocześnie różnica pomiędzy siedmioma i czternastoma dniami nie powinna być aż tak znacząca.

Obliczanie samej wartości błędu może być oparte tylko o samą liczbę nowych przypadków, jednak można to rozszerzyć o wyzdrowienia oraz śmierci. Dzięki temu otrzymane wartości parametrów powinny lepiej odzwierciedlać rzeczywistość. W celu uwzględnienia wielu kryteriów można się posłużyć algorytmem optymalizacji wielokryterialnej lub wykorzystać łączoną miarę błędu. Przykładowym połączeniem błędów jest suma ważona. Przewiduje się, że miara łącząca wszystkie dane najlepiej odda wszystkie parametry, jednak nie musi posiadać najmniejszej wartości funkcji błędu. Zaproponowano sprawdzenie następujących danych rzeczywistych oraz ich połączeń, dzięki którym obliczany jest błąd modelu:

- wykorzystanie nowych wyzdrowień,

- wykorzystanie śmierci,
- łączona miara oparta o zarażenia oraz wyzdrowienia,
- miara łączona wykorzystująca zarażenia, wyzdrowienia i zgony.

Sam proces estymacji warto wykonać z wykorzystaniem różnych zbiorów danych w celu walidacji wyników i uniknięcia wpływu źle dobranych danych, z którymi porównywany jest symulator. W przypadku epidemii COVID-19 można wykorzystać różne zakresy dat, przykładowo „fale”, ale też dane pochodzące z innych krajów. Problemem z wykorzystaniem danych z całego świata jest inny stopień wprowadzanych przez rządy obostrzeń, inne kryteria testowania czy sam wpływ lokalnego klimatu. Ogólnie parametry powinny finalnie otrzymać dość zbliżone wartości, jednak istotne jest sprawdzenie czy na pewno tak się dzieje. W ramach tego badania postanowiono wykorzystać tylko jeden parametr w celu oceny możliwości zastosowania proponowanego modelu do różnych danych. Na podstawie analizy dostępnych danych zdecydowano się na wykorzystanie następujących krajów i dat, które pokrywają się z falami epidemii:

- Polska - 27.09.2020-26.11.2020,
- Polska - 10.03.2020-20.04.2020,
- Czechy - 10.03.2020-20.04.2020,
- Austria - 10.03.2020-20.04.2020.

Na ostatnim etapie badań postanowiono wykorzystać wiedzę zdobytą w ramach wymienionych wcześniej badań i wykorzystać najbardziej obiecujące podejścia, algorytmy do końcowej estymacji parametrów związanych z epidemią COVID-19. Otrzymane w ramach tego badania wartości należy porównać do wartości rzeczywistych, które zostały przedstawione w rozdziale 3.1.



#### 4. BADANIA SYMULACYJNE

W niniejszym rozdziale przedstawiono wyniki badań. Badania wykonano zgodnie z planem przedstawionym w rozdziale 3.5. Na etapie badań wstępnych brane pod uwagę są tylko nowe zarażenia i z tego powodu wartości parametru  $\gamma$  mogą znacznie odstawać od rzeczywistości. Ważne jest wytypowanie jednak rozwiązań bardziej obiecujących, a nie koniecznie dających dobre wyniki, co zostanie sprawdzone dopiero na etapie walidacji. Obliczenia były wykonywane na urządzeniu o następującej specyfikacji:

- procesor - Intel core i5-3470 o czterech rdzeniach i takiej samej ilości wątków taktowanych zegarem 3,2GHz,
- RAM - 8GB jednokanałowej pamięci RAM o taktowaniu 1600Hz,
- system operacyjny - Manjaro Linux 21.0,
- implementacja funkcji BLAS<sup>1</sup> - OpenBLAS.

W każdym przypadku postanowiono przeznaczyć maksymalnie 12 godzin na pojedyncze uruchomienie programów optymalizujących wartości parametrów. W przypadku optymalizatorów umożliwiających zakończenie działania po określonym czasie istnieje możliwość nieznacznie dłuższej pracy ze względu na konieczność dokończenia ostatniej iteracji. Niestety nie wszystkie metody pozwalają na takie działanie i w ich przypadku postanowiono ograniczyć jedynie liczbę iteracji do wartości, która pozwoli na działanie około 12 godzin. Pożądana liczba iteracji została obliczona na podstawie uśrednionego czasu wykonania kilku iteracji z pominięciem pierwszej ze względu na kompilację i cache'owanie. Pierwsza iteracja zajmuje zazwyczaj od 2 do 3 razy dłużej niż następne.

Poszukiwanie wartości dla parametrów  $\sigma$  oraz  $\gamma$  postanowiono przeprowadzić na ich odwrotnościach, czyli średnich liczbach dni wymaganych na przejście do następnego stanu. W ten sposób przestrzeń poszukiwań uległa linearyzacji i odległości pomiędzy kolejnymi wartościami dniowymi są jednakowe. Bezpośrednio działając na parametrach  $\sigma$  i  $\gamma$  dochodzi do znacznego zwiększenia zagęszczenia ilości dni przy wzrastających ich ilościach. W ten sposób przykładowo różnica pomiędzy jednym, a dwoma dniami to wartości z zakresu od 1 do 0,5, czyli połowa całego zakresu możliwych wartości parametrów. Jednocześnie różnica pomiędzy 50 i 100 dniami to już tylko 0,01. Z tego powodu eksploracja przestrzeni w zakresie 0,5 – 1 nie ma niemal znaczenia, a dla wartości bliższych zeru niewielka zmiana parametrów może już znacząco wpłynąć na wynik symulacji.

---

<sup>1</sup> BLAS - Basic Linear Algebra Subprograms

Dla metod wymagających podania punktu początkowego postanowiono ustalić go na wartości wszystkich parametrów równe 0, 1. Pomimo tego, że wszystkie parametry przyjmują wartości z przedziału od 0 do 1, postanowiono zmodyfikować ograniczenia na bardziej racjonalne i pozwalające znacznie zawęzić przestrzeń poszukiwań. Nowe ograniczenia mieszczą w sobie wszystkie możliwe przypadki i mają marginalny wpływ na ukierunkowanie estymacji do poprawnych wyników. Lista zastosowanych ograniczeń przedstawia się następująco:

- $\beta$  - brak zmian,
- $\sigma$  - ograniczenie do zakresu  $[0, 0.25, 1]$ , co przekłada się na możliwy czas przebywania agenta w tym stanie na od 1 do 40 dni,
- $\gamma$  - ograniczenie do zakresu  $[0, 0.25, 1]$ , co przekłada się na możliwy czas przebywania agenta w tym stanie na od 1 do 40 dni.

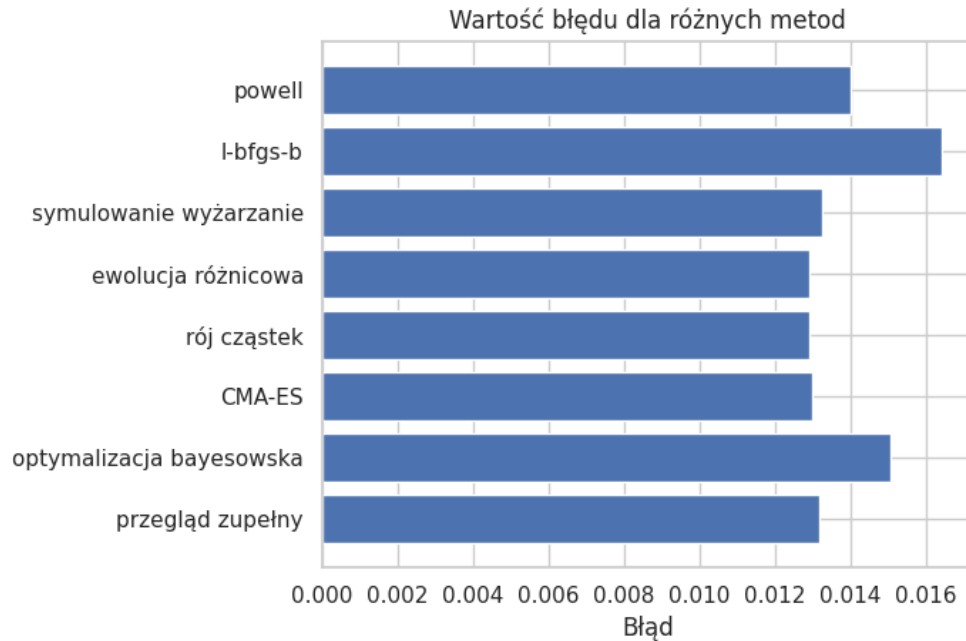
#### 4.1. BADANIA WSTĘPNE

Pierwszym z zaproponowanych badań było sprawdzenie wydajności w zależności od wykorzystanego środowiska uruchomieniowego języka Python. Celem tego badania było porównanie wydajności środowiska CPython, które jest najczęściej używaną implementacją języka Python, oraz środowiska PyPy. Przewidywano co najmniej dwukrotny wzrost wydajności na korzyść PyPy. W przypadku CPython został wykorzystany język Python w najnowszej wersji stabilnej, czyli wersji 3.9. W przypadku środowiska PyPy użyty został język Python w wersji 3.7, co jest najnowszą wersją wspieraną przez to środowisko. Badanie przeprowadzono na algorytmie ewolucji różnicowej (DE) z biblioteki Scipy, z wykorzystaniem algorytmu optymalizacji bayesowskiej (BO) oraz na podstawie wielokrotnych wykonania metody Powella. Wyniki przedstawiono w tabeli 4.1. Czasy podano w minutach oraz sekundach i nie brano pod uwagę pierwszej iteracji ze względu na wyraźnie odstający czas jej wykonania. Z tego powodu czas trwania pierwszej iteracji został przedstawiony w osobnej kolumnie. Ewolucja różnicowa przebiega 2,7 raza szybciej przy pomocy interpretera PyPy, dla metody Powella jest to przyśpieszenie na poziomie 3. Inaczej sytuacja wygląda dla algorytmu optymalizacji bayesowskiej. W tym przypadku standardowa implementacja języka Python radzi sobie znacznie lepiej. Jest to spowodowane wyraźnie wolniejszym działaniem procesu gaussowskiego, odbywającego się w ramach tej procedury, w środowisku PyPy. Sama pierwsza iteracja przebiega jednak szybciej, ponieważ składa się ona jedynie z kilku obliczeń wartości funkcji celu. Interpreter PyPy znacznie lepiej sobie radzi z wykonywaniem symulacji agentowej względem standardowego interpretera CPython i z tego powodu zostanie wykorzystany w dalszych badaniach.

Badanie	Średnia	Mediana	Min	Maks	Pierwsza iteracja
DE - CPython	09:59	09:36	09:05	11:09	23:14
DE - PyPy	03:38	03:36	03:27	03:49	08:26
BO - CPython	00:06.60	00:06.33	00:05.71	00:07.87	12:09
BO - PyPy	00:20.08	00:20.52	00:15.64	00:25.29	04:31
Powell - CPython	08:09	07:39	04:32	15:52	-
Powell - PyPy	02:47	02:31	01:59	06:28	-

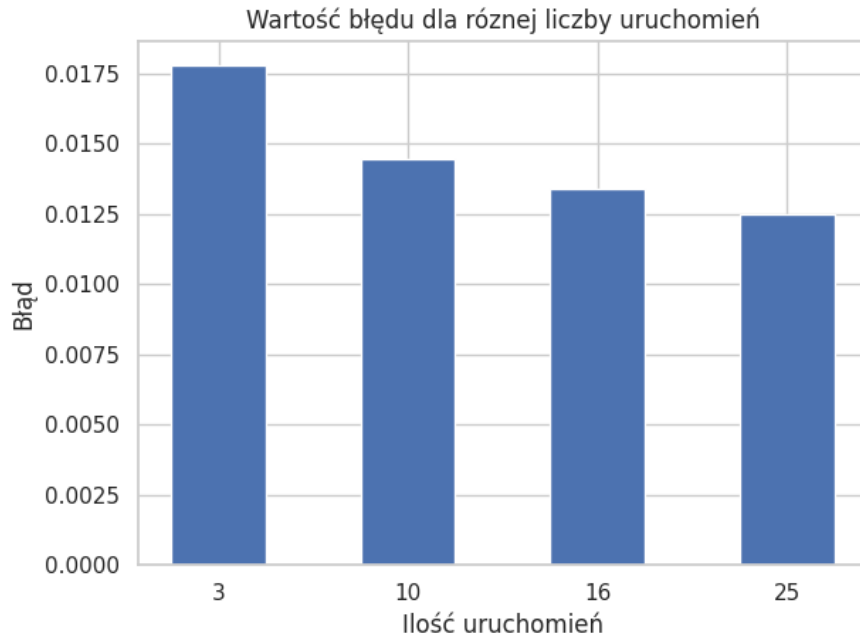
Tabela 4.1: Wyniki badania wydajności dla trzech zaproponowanych algorytmów.

Kolejnym proponowanym badaniem było sprawdzenie działania różnych optymalizatorów w celu wytypowania najbardziej obiecującego z nich. Postanowiono przetestować 8 różnych algorytmów. Nie zostały postawione żadne założenia dotyczące tego, który okaże się najsukcesowniejszy. Wyniki z najmniejszymi osiągniętymi wartościami funkcji błędu dla każdej metody przedstawiano na rysunku 4.1. Metody Powell oraz L-BFGS-B działają lokalnie i z tego powodu zostały uruchomione dla 100 różnych punktów startowych w regularny sposób pokrywających możliwą przestrzeń rozwiązań i nie osiągnęły zbyt niskich wartości błędu. Symulowanie wyżarzanie zostało uruchomione z wartością początkową 0, 1 dla wszystkich parametrów. Metoda optymalizacji bayesowskiej zajęła przedostatnie miejsce pod względem osiągniętej wartości błędu. Metoda przeglądu zupełnego zgodnie z założonym maksymalnym czasem obliczeń wykonała obliczenia na aż 17340 różnych kombinacjach parametrów z regularnym pokryciem możliwej przestrzeni rozwiązań i osiągnęła mniejsze wartości błędu niż wspomniane wcześniej algorytmy. Najmniejsze wartości funkcji błędu zostały osiągnięte dla metody ewolucji różnicowej, roju cząstek oraz CMA-ES. Różnice pomiędzy nimi były bardzo małe i można uznać, że są sobie równe. Ze względu jednak na wygodę użycia oraz prostotę zapisywania progresu procesu optymalizacji w dalszych badaniach postanowiono wykorzystać metodę ewolucji różnicowej.



Rys. 4.1: Wartości błędu osiągnięte przez poszczególne metody optymalizacji.

Celem następnego badania było sprawdzenie wpływu ilości wykonanych symulacji na wartość funkcji błędu. Przewidywane było, że większa liczba uruchomień pozytywnie wpłynie na zmniejszenie jej wartości oraz na sam proces optymalizacji. Wartość funkcji błędu jest obliczana na podstawie danych rzeczywistych oraz uśrednionego przebiegu symulacji z wielu uruchomień. Na rysunku 4.2 przedstawiono wartości błędu osiągnięte w ramach tego badania. W każdym przypadku wykonywane było tyle samo iteracji metody optymalizującej. Uśrednianie z większej liczby wykonanych symulacji zmniejsza wartość błędu i nie ma żadnego wpływu na wartości parametrów. Z przebadanych ilości uruchomień najlepsze wyniki osiągnięto dla wartości 25, jednak różnica względem poprzedniego wyniku nie jest już tak znacząca jak to ma miejsce pomiędzy 3 i 10. Wadą większej liczby uruchomień jest zwiększenie czasu obliczeń potrzebnego na osiągnięcie określonej liczby iteracji algorytmu. Z tego powodu dopuszczalne może być zastosowanie jednej z mniejszych wartości, szczególnie w sytuacji gdzie obecne są ograniczenia czasowe lub ograniczenia w zasobach obliczeniowych, sprzętowych. Na podstawie takiej analizy z dwoma kryteriami postanowiono dokonać ostatecznej ewaluacji z wykorzystaniem 16 wykonanych symulacji, a nie z 25.



Rys. 4.2: Wartości błędu osiągnięte dla uśredniania z wykorzystaniem różnych ilości uruchomień symulacji.

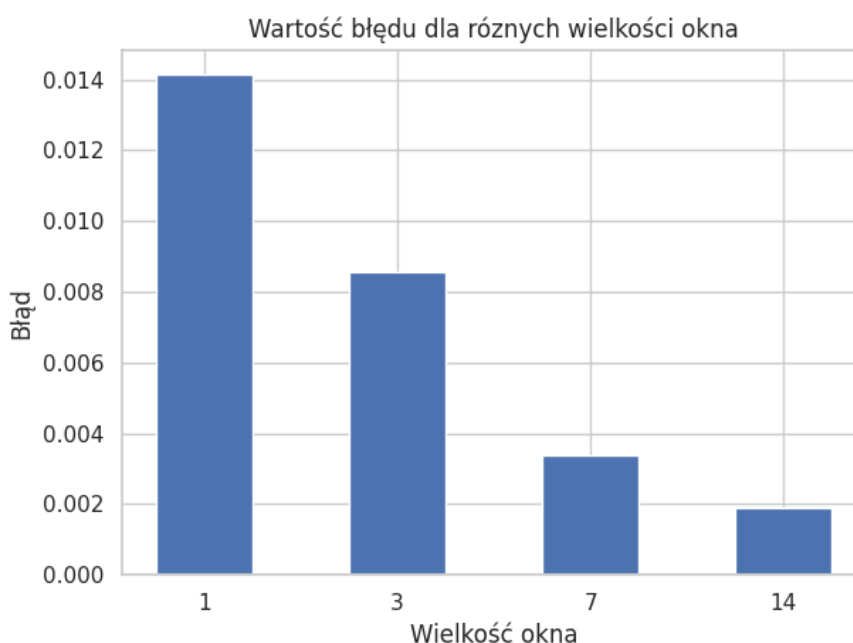
Porównując dane rzeczywiste oraz te generowane przez model, czy to w zastosowaniach estymacji czy uczenia maszynowego, można wykorzystać wiele zróżnicowanych miar i z tego powodu zostało zaproponowane badanie wpływu kilku funkcji błędu. Przewidywane było, że nie będą mieć one żadnego wpływu na otrzymywane wartości parametrów, a sama wartość błędu nie jest istotna ze względu na różne sposoby obliczeń. Błąd był obliczany wyłącznie z wykorzystaniem danych dotyczących nowych zarażeń i z tego powodu wartość  $\gamma$  nie podlegała w żaden sposób optymalizacji. Wyniki zebrano w tabeli 4.2. Wartości parametru  $\beta$  podobnie jak przy poprzednich badaniach wahają się w zakresie od 0,5 do 1,0. Parametr  $\sigma$  dla każdej funkcji błędu przyjmuje zbliżone do siebie wartości. Na podstawie takich przesłanek należy sądzić, że dobór funkcji błędu nie ma wpływu na osiąganе wyniki, co zgadza się z założeniami dotyczącymi tego badania.

Funkcja błędu	Błąd	$\beta$	$1/\sigma$	$1/\gamma$
MSE	0,00036	0,70	10,04	32,17
MAE	0,01413	0,95	8,64	15,07
MedAE	0,00752	0,56	7,89	38,05
MSLE	0,00030	0,80	9,66	22,30

Tabela 4.2: Wyniki badania zastosowanej punkcji błędu na wartości parametrów..

W ramach prac badawczych postanowiono przebadać wpływ wygładzania danych rzeczywistych oraz wyników symulacji. Wygładzanie zaproponowano wykonać z różnymi

rozmiarami okna przesuwnego z wagami jednorodnymi. Przebadane rozmiary okna obejmujące dni to 1, 3, 7, 14. Przewidywano, że zastosowanie uśredniania zmniejszy wartość błędu poprzez usunięcie nagłych zmian wartości oraz tygodniowego trendu. Wyniki dotyczące tego badania przedstawiono na rysunku 4.3. Zastosowanie wygładzania danych oknem przesuwnym ma pozytywny wpływ na wartość błędu. Im większy rozmiar okna tym mniejsza wartość funkcji błędu, a oznacza to lepsze dopasowanie wyników symulacji generowanych przez model do danych. Wyraźnym zwycięzcą w tym badaniu jest uśrednianie oknem przesuwnym o rozmiarze czternastu dni.



Rys. 4.3: Wartości błędu osiągnięte dla wygładzania z wykorzystaniem różnych wielkości okien czasowych.

W przypadku estymacji parametrów epidemii COVID-19 istotne jest wykorzystanie wielu dostępnych danych, a nie tylko dziennej liczby zarażonych jak to było wykonywane w ramach wcześniejszych badań. Błąd modelu względem danych rzeczywistych można obliczać z wykorzystaniem wspomnianej już dziennej liczby wykrytych nowych przypadków, wyzdrowień oraz zgonów. Z tego powodu postanowiono przebadать pięć kombinacji wykorzystywanych danych z wszystkich siedmiu możliwych. Wyniki zostały przedstawione w tabeli 4.3, a ich opisy są następujące:

- **dzienne nowe przypadki** - jest to zbiór wykorzystywany dotychczas jako jedyny we wszystkich badaniach. W tym przypadku wartości parametru  $\gamma$  nie powinny podlegać optymalizacji ponieważ nie mają wpływu na wartość funkcji błędu,
- **dzienne wyzdrowienia** - w tym przypadku algorytm musi dokonać optymalizacji wszystkich parametrów,

- dzienna liczba zgonów - w tym przypadku algorytm musi dokonać optymalizacji wszystkich parametrów, jednak wartości liczbowe zgonów są znacznie mniejsze niż pozostałych rozważanych danych i z tego powodu podejrzewa się mniejszą ich dokładność,
- dzienne nowe przypadki i wyzdrowienia - w takim podejściu optymalizowane są wszystkie parametry i muszą być zachowane odpowiednie zależności pomiędzy nimi ze względu na wykorzystanie kryterium łączonego. Łączny błąd jest sumą obu błędów składowych z wagami równymi 1,
- dzienna liczba przypadków, wyzdrowień oraz zgonów - najbardziej rozbudowana wersja badania wykorzystująca wszystkie rozważane w proponowanym modelu dane. Łączny błąd jest obliczany jako suma ważona błędów składowych. Błąd danych dotyczących nowych przypadków oraz wyzdrowień otrzymały wagę równą 1, a zgony wagę wynoszącą 0,5.

Błędy dla miar łączonych są większe niż w przypadku zsumowania osobno poszczególnych błędów. Dodatkowo nie zaobserwowano większego wpływu ich zastosowania na wartości parametrów. Może to być spowodowane trudniejszym dopasowaniem wyników pracy symulatora do danych rzeczywistych wykorzystując wiele wektorów danych.

Zbiór danych	Błąd	$\beta$	$1/\sigma$	$1/\gamma$
Nowe przypadki	0,01413	0,95	8,64	15,07
Wyzdrowienia	0,01424	0,83	13,57	13,50
Zgony	0,04581	0,64	7,09	10,66
Nowe przypadki i wyzdrowienia	0,03291	0,93	9,75	30,92
Wszystkie 3 wektory	0,09187	0,97	9,27	19,49

Tabela 4.3: Wyniki badania wykorzystania różnych wektorów danych.

W dotychczasowych badaniach wykorzystywane były dane dotyczące drugiej fali w Polsce. Ze względu na globalny zasięg COVID-19 można wykorzystać dane pochodzące z innych krajów oraz różnych okresów. W celu sprawdzenia wpływu wybranego zbioru danych postanowiono wykorzystać dodatkowo dane pochodzące z pierwszej fali w Polsce, Czechach i Austrii. Wyniki przedstawiono w tabeli 4.4. Dla wszystkich wybranych danych otrzymano zbliżone do siebie wyniki, co świadczy o powtarzalności, podobieństwie epidemii w poszczególnych krajach oraz o możliwości zastosowania przedstawionego modelu do różnych sytuacji.

Kraj	Błąd	$\beta$	$1/\sigma$	$1/\gamma$
Polska 2. fala	0,01413	0,95	8,64	15,07
Polska 1. fala	0,02007	0,91	6,90	30,35
Czechy 1. fala	0,03359	0,82	8,63	26,74
Austria 1. fala	0,03602	0,85	11,86	25,52

Tabela 4.4: Wyniki badania wykorzystania różnych krajów i dat.

## 4.2. EWALUACJA I WERYFIKACJA OTRZYMANYCH WARTOŚCI

W niniejszym badaniu przedstawiono porównanie otrzymanych parametrów pandemii COVID-19 w wyniku symulacji z parametrami opracowanymi na podstawie medycznych analiz poszczególnych przypadków. Wartości parametrów wraz ze źródłami zostały przedstawione w rozdziale 3.1 i posłużyły do sprawdzenia poprawności otrzymanych wyników. Wyniki zostały wygenerowane z wykorzystaniem najbardziej obiecujących metod i podejść przebadanych w rozdziale 4.1. Przedstawiają się one następująco:

- środowisko wykonawcze - PyPy,
- metoda optymalizacji - ewolucja różnicowa,
- ilość uruchomień - 16. Ze względów na czas obliczeń nie podjęto decyzji o wykorzystaniu 25 uruchomień,
- funkcja błędu - średni błąd bezwzględny. Wykorzystana miara błędu nie ma wpływu na wartości parametrów,
- transformacja danych - uśrednianie funkcją okna o rozmiarze 14 dni,
- wykorzystane wektory wejściowe - postanowiono wykorzystać wszystkie dostępne w modelu dane,
- wybrane zbiory danych - postanowiono wykorzystać cztery różne zbiory danych,
- ilość uruchomień - 8. Wyniki dla każdego zbioru danych są obliczane na podstawie ośmiu niezależnych wykonania całego procesu optymalizacji,
- 140 iteracji metody optymalizującej.

Badanie to ze względu na wielokrotne powtarzanie z wykorzystaniem wielu zbiorów danych wymagało dużego nakładu obliczeniowego w celu sfinalizowania go. Wyniki wraz z ich niepewnościami zebrano w tabeli 4.5. Najlepszą wartość dopasowania modelu do danych rzeczywistych osiągnięto dla obu przebadanych zakresów dat obejmujących Polskę. W każdym przypadku wartości parametru  $\beta$  oscylują wokół 0,8, co świadczy o powtarzalności tego parametru w różnych krajach i środowiskach. Podobna sytuacja dotyczy parametru  $\gamma$ . W każdym przypadku wartości średnie są do siebie zbliżone, jednak są znacznie większe niż wartości pochodzące z literatury. Przeanalizowana literatura wskazuje, iż  $\gamma$  powinna znajdować się w zakresie od 8 do 14, a w skrajnym przypadku



do 20 dni. W ramach tego badania jednak otrzymał on wartość około 34,5 dnia, co może wskazywać na nieuwzględnienie pewnych szczegółów w proponowanym modelu lub na sam problem zastosowanej metody estymacji. Istotnym czynnikiem jest również sama niepewność danych i sposobu w jaki były one raportowane. W przypadku Polski wartości parametru  $\sigma$  znajdują się w zakresach proponowanych w literaturze, jednak są nieznacznie większe niż sugerowane wartości średnie. Znacznie inaczej jest w przypadku pozostałych wybranych danych, czyli dla Czech oraz Austrii. W tych przypadkach wartości są znacznie większe oraz charakteryzują się dużą niepewnością, rozrzutem wartości wskazywanym przez odchylenie standardowe. Otrzymane wynikowe wartości parametru  $\sigma$  tylko częściowo pokrywają się z tymi pochodzącymi z literatury.

Kraj	Błąd	$\beta$	$1/\sigma$	$1/\gamma$
Polska 2. fala	0.01351 (0.00090)	0.76 (0.21)	9.30 (1.37)	33.76 (4.17)
Polska 1. fala	0.02901 (0.00073)	0.86 (0.09)	5.48 (0.71)	34.17 (5.39)
Czechy 1. fala	0.05674 (0.00317)	0.76 (0.20)	21.69 (12.42)	35.51 (3.42)
Austria 1. fala	0.04794 (0.00508)	0.79 (0.09)	28.49 (5.51)	35.42 (3.26)

Tabela 4.5: Wyniki badania. Podano wartości średnie z ośmiu uruchomień optymalizacji, a w nawiasach zostało przedstawione odchylenie standardowe.

## **ZAKOŃCZENIE**

### **PERSPEKTYWY ROZWOJOWE**

Założony cel pracy został osiągnięty. Praca została przygotowana w zakresie zdefiniowanym na początku, jednak istnieje wciąż wiele możliwości na jej rozszerzenie. Jednym z możliwych kierunków jest wykorzystanie innych form i rozszerzeń modelu agentowego czy użycie opisu czysto matematycznego. W ramach pracy agencji poruszali się w środowisku regularnej siatki, lecz możliwe jest zastosowanie grafu. Taki graf może tworzyć sieć miast i miejscowości lub sieć społeczną, gdzie każdy agent jest pojedynczym węzłem, a krawędzie oznaczają kontakty. Wciąż istnieje szerokie pole do zastosowania bardziej złożonych modeli i stanów. W wielu cytowanych pracach dochodziło do takich rozszerzeń. Definicja samych zachowań agentów również może ulec zmianie i uwzględniać przemieszczanie się zależne od punktów zainteresowań lub kontaktów. Taki agent częściej by poruszał się do pewnych miejsc lub w kierunku znanych sobie agentów, jak ma to miejsce w rzeczywistości. Kolejnym krokiem może być rozszerzenie badań. W samych badaniach wydajności można przetestować również inne środowiska wykonawcze, sprawdzić wpływ optymalizacji w kompilacji języka Python, która między innymi wyłącza asercje. Innym kierunkiem zwiększania wydajności jest manipulacja parametrami pythonowego garbage collector. W trakcie badań procesor był w pełni wykorzystany, jednak zużycie pamięci dochodziło maksymalnie do 200MB. Można sprawdzić wpływ rzadszego uruchamiania GC na wydajność programów. Wciąż istnieje wiele metod estymacji i optymalizacji, które nie zostały sprawdzone w ramach tej pracy, a mogłyby uzyskać zadowalające wyniki. Przebadane metody również posiadają wiele parametrów, które można dostosowywać w celu lepszego działania. Proponowany model można też zastosować do innych krajów, a nawet chorób. Powinien on odpowiadać również innym epidemii, takim jak przykładowo SARS.

### **PODSUMOWANIE**

Celem pracy była estymacja parametrów epidemii COVID-19 z wykorzystaniem modelu agentowego. Rozważany był model SEIR, w którym pojedynczy agent może znajdować się w danym czasie w jednym z czterech stanów, a przejścia pomiędzy nimi są dokładnie zdefiniowane.

Niniejsza praca stanowi ogłód sytuacji związanej z pandemią COVID-19. Przedstawiono możliwe sposoby modelowania otaczającego nas świata, wraz z przykładami zastosowań.

Zaprezentowano podstawowe modele wykorzystywane w modelowaniu rozwoju chorób w populacjach oraz stojące za nimi równania różniczkowe. Przeprowadzono dogłębną analizę literatury w celu ustalenia rzeczywistych parametrów epidemiologicznych COVID-19 oraz w celu sprawdzenia już istniejących rozwiązań związanych z modelowaniem i estymacją parametrów tej epidemii. Zaprezentowany został własny model, który posłużył do modelowania COVID-19 w ramach tej pracy. Dokładnie przedstawiono plan badań i spodziewane w ramach nich wyniki. Wyniki przeprowadzonych badań przedstawiono w odpowiednich formach oraz opisano je. Jednym z bardziej zaskakujących wyników jest wartość funkcji błędu osiągnięta przez metodę przeglądu zupełnego. Podejście to okazało się skuteczniejsze niż cztery inne sprawdzone metody optymalizacji. Wyniki estymacji częściowo pokrywają się z parametrami medycznymi, co świadczy o skuteczności proponowanego podejścia do estymacji. Przedstawiony model można rozszerzać i sprawdzać inne metody estymacji parametrów. Istotnym czynnikiem okazał się również dobór danych dotyczących pandemii, który ma bardzo duży wpływ na to czy model jest w stanie oddać wybrane dane. Problemem jaki również wystąpił był czas działania symulacji agentowej. Z tego powodu postanowiono przetestować różne środowiska wykonawcze dla kodu napisanego w języku Python. Dwóch z sprawdzonych środowisk nie udało się uruchomić i nie zostały wykorzystane w tej pracy, jednak pozostałe dwa bezproblemowo działały. Środowisko PyPy okazało się znacznie szybsze od standardowego CPython. Może to być wskazówka dla twórców oprogramowania pisanego w języku Python, iż może okazać się warte wykorzystanie innego środowiska w celu uruchamiania swoich programów.

## BIBLIOGRAFIA

- [1] Backer, J.A., Klinkenberg, D., Wallinga, J., *Incubation period of 2019 novel coronavirus (2019-ncov) infections among travellers from wuhan, china, 20–28 january 2020*, Eurosurveillance. 2020, tom 25, 5, str. 2000062.
- [2] Bill & Melinda Gates Foundation, *Seir and seirs models*, <https://docs.idmod.org/projects/emod-generic/en/latest/model-sir.html>. [Online; Dostęp: 29.04.2021].
- [3] Bill & Melinda Gates Foundation, *Sir and sirs models*, <https://docs.idmod.org/projects/emod-generic/en/latest/model-sir.html>. [Online; Dostęp: 29.04.2021].
- [4] Böhning, D., Rocchetti, I., Maruotti, A., Holling, H., *Estimating the undetected infections in the covid-19 outbreak by harnessing capture–recapture methods*, International Journal of Infectious Diseases. 2020, tom 97, str. 197–201.
- [5] Centers for Disease Control and Prevention, *Interim clinical guidance for management of patients with confirmed coronavirus disease (covid-19)*, <https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-guidance-management-patients.html>. [Online; Dostęp: 20.05.2021].
- [6] Dong, E., Du, H., Gardner, L., *An interactive web-based dashboard to track covid-19 in real time*, The Lancet infectious diseases. 2020, tom 20, 5, str. 533–534.
- [7] European Centre for Disease Prevention and Control, *Questions and answers on covid-19: Basic facts*, <https://www.ecdc.europa.eu/en/covid-19/questions-answers/questions-answers-basic-facts>. [Online; Dostęp: 20.05.2021].
- [8] Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C., *DEAP: Evolutionary algorithms made easy*, Journal of Machine Learning Research. 2012, tom 13, str. 2171–2175.
- [9] Foryś, U., Poleszczuk, J., *Modele oddziaływań między dwiema populacjami i*, <http://mst.mimuw.edu.pl/lecture.php?lecture=mbm&part=Ch6>. [Online; Dostęp: 29.04.2021].
- [10] Godio, A., Pace, F., Vergnano, A., *Seir modeling of the italian epidemic of sars-cov-2 using computational swarm intelligence*, International Journal of Environmental Research and Public Health. 2020, tom 17, 10.
- [11] Guan, W.j., Ni, Z.y., Hu, Y., Liang, W.h., Ou, C.q., He, J.x., Liu, L., Shan, H., Lei, C.l., Hui, D.S., i in., *Clinical characteristics of coronavirus disease 2019 in china*, New England journal of medicine. 2020, tom 382, 18, str. 1708–1720.
- [12] He, S., Peng, Y., Sun, K., *Seir modeling of the covid-19 and its dynamics*, Nonlinear Dynamics. 2020, tom 101, 3, str. 1667–1680.
- [13] Hoertel, N., Blachier, M., Blanco, C., Olfson, M., Massetti, M., Rico, M.S., Limosin, F., Leleu, H., *A stochastic agent-based model of the sars-cov-2 epidemic in france*, Nature medicine. 2020, tom 26, 9, str. 1417–1421.
- [14] Kazil, J., Masad, D., Crooks, A., *Utilizing python for agent-based modeling: The mesa*

- framework, w: Social, Cultural, and Behavioral Modeling*, pod red. R. Thomson, H. Bisgin, C. Dancy, A. Hyder, M. Hussain (Springer International Publishing, Cham, 2020), str. 308–317.
- [15] Kermack, W.O., McKendrick, A.G., *A contribution to the mathematical theory of epidemics*, Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character. 1927, tom 115, 772, str. 700–721.
  - [16] Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., Ren, R., Leung, K.S., Lau, E.H., Wong, J.Y., i in., *Early transmission dynamics in wuhan, china, of novel coronavirus–infected pneumonia*, New England journal of medicine. 2020.
  - [17] Max Roser, Hannah Ritchie, E.O.O., Hasell, J., *Coronavirus pandemic (covid-19)*, Our World in Data. 2020. <https://ourworldindata.org/coronavirus>.
  - [18] McAloon, C., Collins, Á., Hunt, K., Barber, A., Byrne, A.W., Butler, F., Casey, M., Griffin, J., Lane, E., McEvoy, D., i in., *Incubation period of covid-19: a rapid systematic review and meta-analysis of observational research*, BMJ open. 2020, tom 10, 8, str. e039652.
  - [19] Miranda, L.J.V., *PySwarms, a research-toolkit for Particle Swarm Optimization in Python*, Journal of Open Source Software. 2018, tom 3.
  - [20] Mizumoto, K., Kagaya, K., Zarebski, A., Chowell, G., *Estimating the asymptomatic proportion of coronavirus disease 2019 (covid-19) cases on board the diamond princess cruise ship, yokohama, japan, 2020*, Eurosurveillance. 2020, tom 25.
  - [21] scikit optimize, *scikit-optimize*, <https://scikit-optimize.github.io/stable/>. [Online; Dostęp: 24.05.2021].
  - [22] Organizacja Narodów Zjednoczonych, *Demographic yearbook – 2019*, <https://unstats.un.org/unsd/demographic-social/products/dyb/documents/dyb2019/table07.pdf>. [Online; Dostęp: 22.05.2021].
  - [23] Reynolds, C.W., *Flocks, herds and schools: A distributed behavioral model*, w: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), str. 25–34.
  - [24] Verity, R., Okell, L.C., Dorigatti, I., Winskill, P., Whittaker, C., Imai, N., Cuomo-Dannenburg, G., Thompson, H., Walker, P.G., Fu, H., i in., *Estimates of the severity of coronavirus disease 2019: a model-based analysis*, The Lancet infectious diseases. 2020, tom 20, 6, str. 669–677.
  - [25] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burrows, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods. 2020, tom 17, str. 261–272.
  - [26] Volterra, V., *Variations and fluctuations of the number of individuals in animal species living together*, ICES Journal of Marine Science. 1928, tom 3, 1, str. 3–51.
  - [27] Wan, K., Chen, J., Lu, C., Dong, L., Wu, Z., Zhang, L., *When will the battle against novel*

*coronavirus end in wuhan: A seir modeling analysis*, Journal of global health. 2020, tom 10, 1.

- [28] Wu, S.L., Mertens, A.N., Crider, Y.S., Nguyen, A., Pokpongkiet, N.N., Djajadi, S., Seth, A., Hsiang, M.S., Colford, J.M., Reingold, A., i in., *Substantial underestimation of sars-cov-2 infection in the united states*, Nature communications. 2020, tom 11, 1, str. 1–10.
- [29] Wu, Z., McGoogan, J.M., *Characteristics of and important lessons from the coronavirus disease 2019 (covid-19) outbreak in china: summary of a report of 72 314 cases from the chinese center for disease control and prevention*, Jama. 2020, tom 323, 13, str. 1239–1242.
- [30] Yang, Z., Zeng, Z., Wang, K., Wong, S.S., Liang, W., Zanin, M., Liu, P., Cao, X., Gao, Z., Mai, Z., i in., *Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions*, Journal of thoracic disease. 2020, tom 12, 3, str. 165.

## SPIS RYSUNKÓW

2.1	Wykres wielkości populacji w modelu Lotka-Volterra. Źródło [9] . . . . .	7
2.2	Model Lotki-Volterry w wersji agentowej. . . . .	8
2.3	Rozwój prostego automatu komórkowego. (Opracowanie własne) . . . . .	9
2.4	Stany w modelu SIR. Opracowano na podstawie [3]. . . . .	10
2.5	Stany w modelu SEIR. Opracowano na podstawie [2]. . . . .	11
2.6	Przykłady dostępnych przestrzeni. . . . .	14
2.7	Przykładowy widok okna Mesa. Można wyróżnić: widok symulacji, generowane wykresy oraz kontrolki. . . . .	15
3.1	Stany i przejścia pomiędzy nimi w proponowanym modelu. . . . .	23
3.2	Przykładowy przebieg symulacji z wykorzystaniem środowiska Mesa. . . . .	24
3.3	Wpływ ilości uruchomień na powtarzalność wyników. Uśrednianie z wykorzystaniem większej liczby uruchomień zmniejsza rozrzut generowanych przebiegów. . . . .	28
4.1	Wartości błędu osiągnięte przez poszczególne metody optymalizacji. . . . .	34
4.2	Wartości błędu osiągnięte dla uśredniania z wykorzystaniem różnych ilości uruchomień symulacji. . . . .	35
4.3	Wartości błędu osiągnięte dla wygładzania z wykorzystaniem różnych wielkości okien czasowych. . . . .	36