

Modele rozwoju epidemii

Tomasz Ś.

Tobiasz P.

Kraków, 2020

Naukowcy od wielu lat studiują przypadki epidemii występujących na świecie. W samym ostatnim dwudziestoleciu świat zmagał się z wirusami takimi jak: SARS, MERS czy Ebola. Dzisiejsza sytuacja jest bardzo wyjątkowa, bo o ile epidemie występują stosunkowo często to pandemie, globalne zarazy, należą raczej do rzadkości. Ich pojawienie ma wpływ na wszystkie aspekty życia na całym globie, dlatego też są wdzięcznym tematem badań dla uczonych z wielu dziedzin. Według nas walkę z epidemią można podzielić na dwa główne działy - medyczny oraz społeczny. Wysiłki personelu medycznego skupiają się na pomocy już zarażonym, lecz kluczowe w tego typu sytuacjach jest zatrzymanie dalszego rozwoju zarazy. Kwestie biologiczne takie jak budowa patogenu, jego wpływ na organizm czy sposób leczenia to jedna strona medalu, istotna jest również strategia zarządzania społeczeństwem w taki sposób, aby zarażeń było jak najmniej. Przydatnymi narzędziami do planowania kolejnych ruchów w walce z zarazą są modele matematyczne rozwoju epidemii. Głównym ich zadaniem jest przewidywanie dynamiki rozprzestrzeniania się choroby oraz wpływu na nią wprowadzanych obostrzeń. Historia ich powstawania sięga początków XX wieku. Jednym z pierwszych, a zarazem najprostszym, jest model SIR (S - Susceptible, I - Infectious, R - Recoverd).

1 SIRD

Jest to model bazujący na modelu SIR. Oprócz osób zdrowych - S, zakażonych - I, wyzdrowiałych - R, bierze pod uwagę śmiertelność osób zakażonych - D. Składa się z układu czterech równań różniczkowych pierwszego rzędu o następującej postaci:

$$\begin{aligned}\dot{S} &= -b\frac{SI}{N} \\ \dot{I} &= b\frac{SI}{N} - kI - mI \\ \dot{R} &= kI \\ \dot{D} &= mI\end{aligned}$$

gdzie: b - współczynnik zaraźliwości; k - współczynnik wyzdrowień; m - współczynnik śmiertelności; S - liczba osób możliwych do zakażenia; I - liczba osób zakażonych; R - liczba osób wyzdrowiałych; D - liczba osób zmarłych; $N = S + I + R + D$;

- Współczynnik zaraźliwości - b

Jest to istotny element do zrozumienia modelu. Oznacza on ile osób w ciągu jednej jednostki czasu (dnia) jest wstanie zarazić zakażony. Może on być różnie definiowany, w naszym przypadku:

$$b = avgC * tP$$

gdzie:

avgC - średnia ilość kontaktów osoby zakażonej na jednostkę czasu
tP - prawdopodobieństwo transmisji patogenu na osobę zdrową

W wielu modela współczynnik ten jest zależny od czasu, można go zmodyfikować tak aby uwzględniał dystansowanie społeczne, wprowadzenie kwarantanny, czy zamknięcie części gospodarki.

- Współczynnik wyzdrowienia - k

Dla nas główne znaczenie ma jego odwrotność. Mianowicie:

$\frac{1}{k}$ - ilość jednostek czasu potrzebnych do wyzdrowienia.

- Współczynnik śmiertelności - m

Bardzo ważna jest dobra estymacja tego współczynnika. Jest to stosunek śmierci spowodowanych chorobą do wszystkich zarażonych.

2. Dodatkowe wskaźniki

Warto uwzględnić w naszych rozważaniach wskaźnik rozprzestrzeniania się choroby:

$$Rt = \frac{b S_0}{k N}$$

gdzie: S_0 to cała populacja pomniejszona o początkową liczbę zarażonych

3. Rozwiązanie

Rozwiązaliśmy powyższy układ równań różniczkowych względem czasu.

Przyjeliśmy następujące warunki początkowe:

- Czas
Naszą podstawową jednostką czasu jest jeden dzień. Rozwiązanie przeprowadziliśmy na przestrzeni 365 dni.
- Współczynnik - b
Założyliśmy, że patogen transportuje się z prawdopodobieństwem równym 4%, a średnia ilość dziennych spotkań wynosi 20. W kolejnych krokach zmienialiśmy liczbę spotkań : 17, 12, 8, 4.
- Współczynnik - k
Założyliśmy, że czas wyleczenia trwa ok. 3 tygodni czyli 21 dni.
- Współczynnik - m
Ustaliliśmy, według doniesień naukowców, że śmiertelność wynosi ok. 0.3%.
- Populacja - N
Przyjeliśmy populację podlegającą epidemi na poziomie 38 000 000 osób.
- Zainfekowani - I
Przyjeliśmy początkową liczbę zainfekowanych na poziomie 500 osób.

```
[1]: import numpy as np
      from numpy import array, arange
      from scipy.integrate import odeint
      import matplotlib.pyplot as plt
```

```
[2]: N=38000000 # Cała populacja
      IO=500     # Liczba startowa zainfekowanych osób
      SO=N-IO    # Liczba startowa osób możliwych do zarażenia

      R0=0 # Liczba startowa osób wyzdrowiałych
      D0=0 # Liczba startowa osób zmarłych

      avgC = 4
      tP = 0.04
      Days = 21 # Dni potrzebne do wyzdrowienia
      m=0.003

      b=avgC*tP
      b1=17*tP
      b2=12*tP
      b3=8*tP
      b4=4*tP
      k=1/Days
```

```
[3]: print("b = ",b)
      print("k = ",k)
      print("m = ",m)
      print("Rt = ",b/k*SO/N)
```

```
b = 0.16
k = 0.047619047619047616
m = 0.003
Rt = 3.3599557894736845
```

```
[4]: y0=[SO,IO,R0,D0]
      t=np.linspace(0,365,1000)
```

```
[5]: def y(p,t,b,k,m):
      S=p[0]
      I=p[1]
      R=p[2]
      f1=-b*S*I/N
      f2=b*S*I/N-k*I-m*I
      f3=k*I
      f4=m*I
      return array([f1,f2,f3,f4])
```

```
[6]: sol = odeint(y,y0,t,args=(b,k,m)) #rozwiązywanie
sol1 = odeint(y,y0,t,args=(b1,k,m))
sol2 = odeint(y,y0,t,args=(b2,k,m))
sol3 = odeint(y,y0,t,args=(b3,k,m))
sol4 = odeint(y,y0,t,args=(b4,k,m))

Rt=[]
for x in sol[:, 0]:
    Rt.append(b*1/k*x/N) # Obliczanie Rt
```

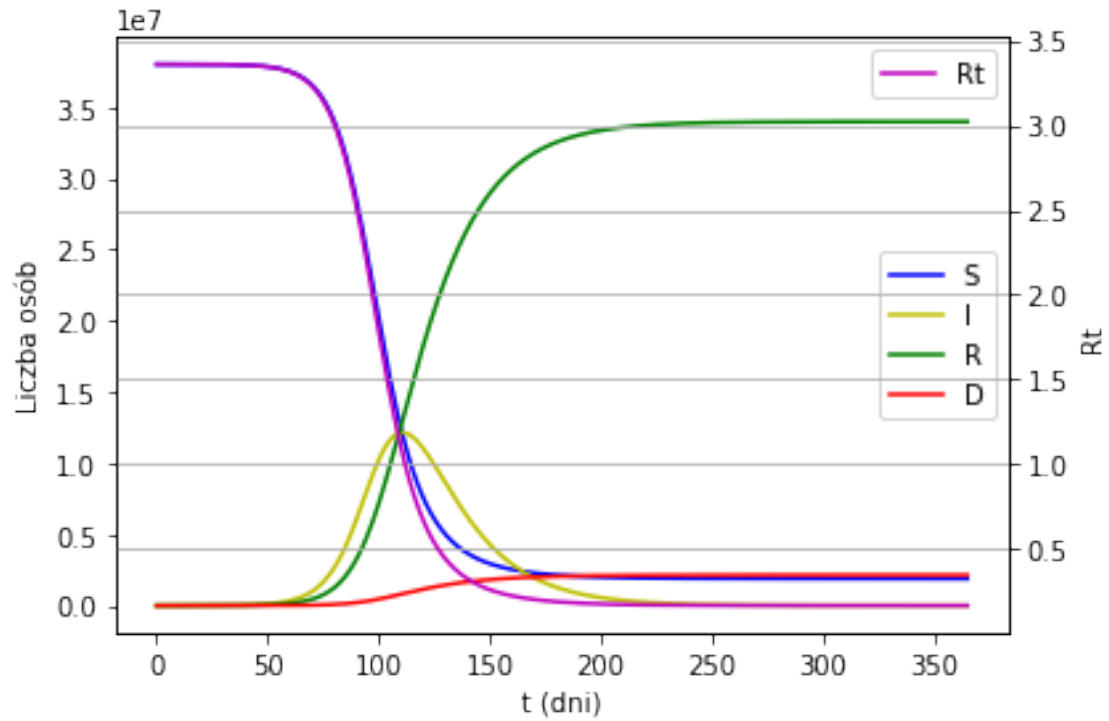
```
[7]: fig, ax1 = plt.subplots()

ax1.set_xlabel('t (dni)')
ax1.set_ylabel('Liczba osób')
ax1.plot(t, sol[:, 0], 'b', label='S')
ax1.plot(t, sol[:, 1], 'y', label='I')
ax1.plot(t, sol[:, 2], 'g', label='R')
ax1.plot(t, sol[:, 3], 'r', label='D')
ax1.legend(loc='best')

ax2 = ax1.twinx()
ax2.set_ylabel('Rt')
ax2.plot(t,Rt, 'm', label='Rt')
ax2.legend(loc='best')

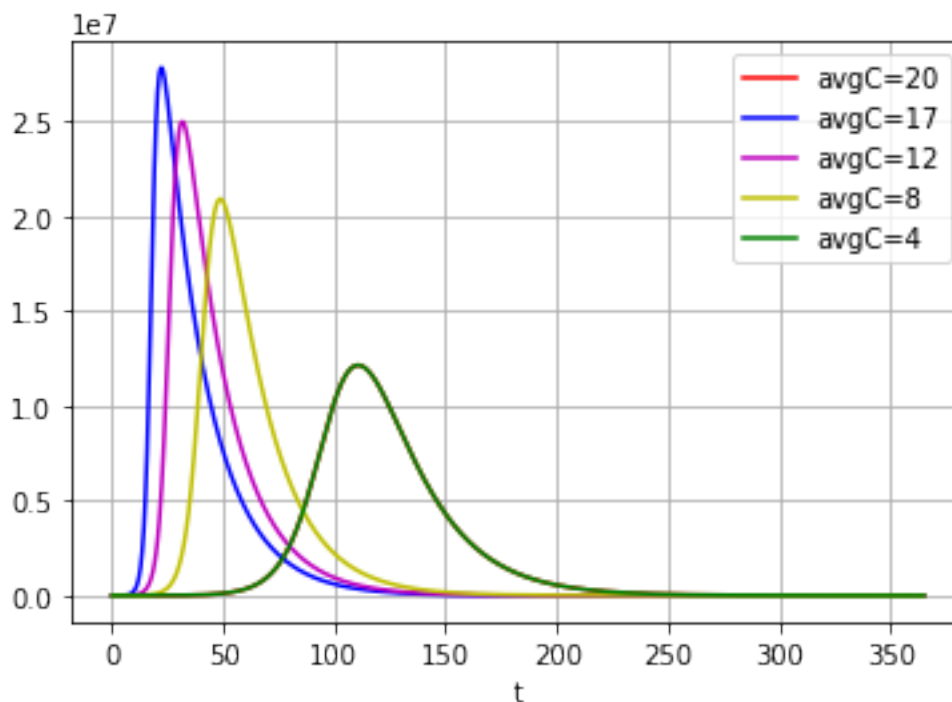
fig.tight_layout()

plt.grid()
plt.show()
```



Rys. 1

```
[8]: plt.plot(t,sol[:,1], 'r', label='avgC=20')
plt.plot(t,sol1[:,1], 'b', label='avgC=17')
plt.plot(t, sol2[:, 1], 'm', label='avgC=12')
plt.plot(t, sol3[:, 1], 'y', label='avgC=8')
plt.plot(t, sol4[:, 1], 'g', label='avgC=4')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```



Rys. 2

4. Wnioski

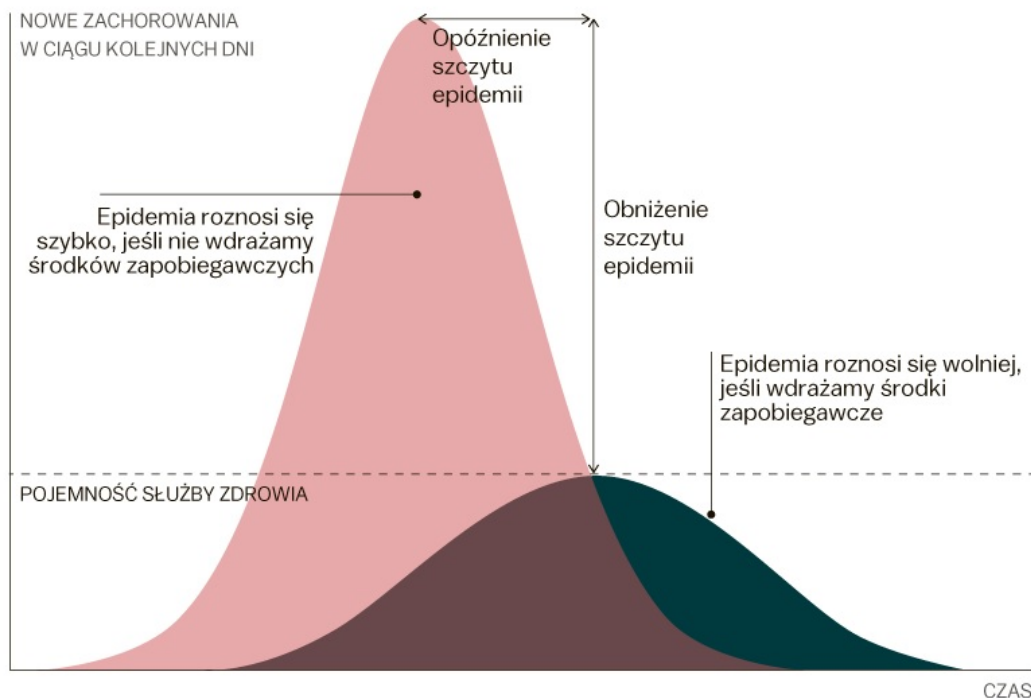
Rozwiązanie w dużej mierze zależy od definicji współczynników zaraźliwości oraz wyzdrowień. W naszej analizie posłużyliśmy się stosunkowo prostą definicją, jednak już ona daje nam możliwość zaprezentowania kilku pojęć istotnych z punktu widzenia zwykłego obywatela i zrozumienia potrzeby wprowadzanych obostrzeń. Wiele było mówione o tak zwanej pojemności służby zdrowia. Przez to pojęcie rozumiemy liczbę osób, którymi szpitale są w stanie się zaopiekować. Rysunek nr 3 obrazuje przyrost liczby zarażonych w czasie oraz granicę, do której służba zdrowia działa sprawnie i obsługuje normalnie pacjentów. Niekontrolowana epidemia rozwija się bardzo szybko, a liczba zarażonych rośnie wykładniczo. W takiej sytuacji odpowiedź ze strony służb medycznych może być niewystarczająca. Istotne jest rozciągnięcie przyrostu zarażonych w czasie, tak aby można było przygotować dodatkową pomoc oraz ewentualnie zastępować wyleczonych nowo zarażonymi. W tym celu, w wielu miejscach na świecie wprowadzono zasady izolacji społecznej. Zmniejszenie liczby kontaktów między ludźmi skutkuje znacznie wolniejszym wzrostem liczby zachorowań. Można to prześledzić zmniejszając parametr $avgC$. Jak pokazuje wykres z rys. nr 2 z mniejszym parametrem $avgC$ krzywa się spłaszcza, a liczba zachorowań przyrasta znacznie wolniej.

Kolejnym ważnym elementem, który należy śledzić jest wartość wskaźnika rozprzestrzeniania się choroby R_t . Mówi nam o liczbie osób zarażanych przez jedną zainfekowaną jednostkę. Jeżeli $R_t < 1$ wówczas mamy do czynienia z cofaniem się epidemii, co można zaobserwować na rys. nr 1. Przy spadku wartości R_t poniżej 1 następuje spadek liczby zakażonych. Współczynnik ten jest proporcjonalny do średniej ilości spotkań. Ponownie, stosując zasady dystansu społecznego

zmniejszamy bezpośrednio zaraźliwość.

Musimy grać na czas. W ten sposób uratujemy tysiące osób

Schemat rozprzestrzeniania się epidemii przy zastosowaniu środków zapobiegawczych i bez ich zastosowania.



© GAZETA WYBORCZA

ŹRÓDŁO: CDC,
OPRACOWANIE: BIQDATA

Rys. 3

2 Stochastyczne modele epidemii

1. Wstęp

Rozprzestrzenianie choroby można również modelować z użyciem metod Monte Carlo. Jednym z modeli, który możemy wykorzystać w naszym problemie jest model perkolacyjny. Pierwotnie dotyczył on rozchodzenia się cieczy w porowatym materiale, jednak my omówimy go sobie na przykładzie płonącego lasu, który działa na zasadzie automatu komórkowego - podlega ewolucji w dyskretnych przedziałach czasu według ściśle określonych reguł.

2. Rozwiązanie

W naszych rozważaniach przyjmiemy obszar w kształcie kwadratu o boku 100 jednostek, czyli sumarycznie uzyskaliśmy 10000 komórek. Następnie rozlokujemy w obszarze drzewa. Liczba drzew jest zależna od parametru p , który możemy określić jako parametr gęstości. Jego maksymalna wartość wynosi jeden i jego iloczyn z liczbą miejsc w obszarze daje liczbę drzew do zasadzenia. Za pomocą generatora liczb pseudolosowych, który losuje liczby z zakresu od 0 do wartości długości boku obszaru o rozkładzie jednostajnym, uzyskujemy parę współrzędnych, które wskazują miejsce zasadzenia drzewa. Jeżeli miejsce jest już zajęte albo należy do brzegu obszaru ponawiamy losowanie. Na koniec zostaje podpalony ostatni rząd, co stanowi źródło pożaru. W komórkach umieszczamy liczby odpowiadające poszczególnym stanom w jakich się ona znajduje:

- 9 - brzeg obszaru
- 0 - pusta komórka
- 1 - komór z drzewem
- 2 - płonące drzewo
- 3 - spalone drzewo

Po określeniu warunków początkowych następuje iteracja po wszystkich komórkach obszaru. W wypadku napotkania płonącego drzewa następuje rozprzestrzenienie ognia na sąsiadujące z nim drzewa. Czyli zakładając, że drzewo ma położenie (x,y) to ogień przenosi się na komórki $(x-1,y)$ oraz $(x,y-1)$, oczywiście jeżeli w tych komórkach nie znajduje się spalone lub płonące drzewo. Należy zaznaczyć, że ogień rozchodzi się w przeciwnym kierunku do przebiegu pętli po obszarze. Wynika to z założenia, że jednorazowy przebieg pętli po całym obszarze to jedna jednostka czasu. Zabieg ten ma na celu wykluczenie sytuacji, w której cały las zostaje podpalony w jednym momencie oraz pozwala rozpatrywać zjawisko w czasie.

Poniżej prezentujemy funkcję przyjmującą parametry p oraz rozmiar obszaru, a zwraca liczbę drzew, liczbę spalonych drzew oraz czas czyli liczbę iteracji po całym obszarze

```
[9]: from random import *

seed(1)
size_x=100
size_y=100
p=0.6
tab = [[0 for i in range(0,size_x)] for j in range(0,size_y)]
```

```
[11]: # 0 - pusta kratka
# 1 - zapełniona kratka
# 2 - płonąca kratka
# 3 - zgaszona kratka
# 9 - brzeg obszaru
from random import *

seed(1)

def pom(i,j,tab):
```



```

        if tab[i-1][j] == 1:
            tab[i-1][j] = 2
        if tab[i][j-1] == 1:
            tab[i][j-1] = 2

def fire(p,size):
    uk_size = size-2
    pk_size = 1
    tab = [[0 for i in range(0,size)] for j in range(0,size)]
    per=False

    for i in range(0,size):
        tab[i][0]=9
        tab[0][i]=9
        tab[size-1][i]=9
        tab[i][size-1]=9

    count_tree = int((size*size-(size*4-4))*p)
    tree_left=count_tree

    while tree_left > 0:
        while True:
            i=randint(pk_size,uk_size)
            j=randint(pk_size,uk_size)
            if tab[i][j] == 1:
                continue
            else:
                tab[i][j] = 1
                tree_left-=1
                break;

    for i in range(pk_size,uk_size):
        if tab[uk_size][i] == 1:
            tab[uk_size][i]=2;
        else:
            continue

    burnt_trees=0
    time =0
    tmp=0

    while True:
        for i in range(pk_size,uk_size+1):
            for j in range(pk_size,uk_size+1):
                if tab[i][j] == 2:

```

```

        pom(i,j,tab)
        tab[i][j] = 3
        burnt_trees+=1
        tmp+=1

    time+=1
    if tmp == 0:
        break
    else :
        tmp=0

for i in range(pk_size,uk_size):
    if tab[1][i]==3:
        per=True
        break

return [count_tree,burnt_trees,time,per]

```

3. Wnioski

Powyższy model prezentuje również rozprzestrzenianie się choroby w społeczeństwie. Poszczególne drzewa to ludzie, drzewa spalone to ludzie wyleczenie lub zmarli, natomiast płonące drzewa to zainfekowani. Parametr perkolacji p można utożsamić z parametrem b z modelu SIR. W tym przypadku podobnie jak z przykładem płonącego lasu jest on parametrem gęstości. Im większa jego wartość tym więcej tworzy się skupisk ludzi, w których jeden osobnik infekuje całą grupę. W najgorszym wypadku wszystkie osobniki są ze sobą połączone co oznacza, że zarazi się całą populacją. Parametr p w tym modelu zawiera w sobie zaraźliwość wirusa, poziom przestrzegania higieny czy dystansu społecznego. Ponownie wiele zależy od tego jak go zdefiniujemy.

Na rysunku nr 4 prezentujemy procentową ilość osobników zarażonych podczas trwania epidemii w zależności od parametru perkolacji. Przy pewnej wartości parametru p można zauważyć nagły wzrost ilości zakażonych.

```

[16]: size=100

p=0.8
t=[]
w=[]

per=[0]*100

for i in np.arange(0.01,1,0.01):
    y=fire(i,size)
    t.append(y[1])
    w.append(y[1]/y[0]*100)
    print("p: ", i, " Trees: ",y[0], " Burnt Trees: ",y[1], " Per: ",y[3], "
    ↳Time: ", y[2])

```

```
[17]: r=np.arange(0.01,1,0.01)
plt.plot(r,w, 'r', label='Zakaženi(p)')
plt.legend(loc='best')
plt.xlabel('p')
plt.grid()
plt.show()
```



Rys. 4