

Projekt Indywidualny NetLogo
Model ABM
rozprzestrzeniania się plotki w zespole

Autor: Kosiorowska Marcelina

Nr indeksu: 268772

Grupa laboratoryjna: Wtorek 7:30

Wstęp

Celem niniejszego projektu była budowa własnego modelu agentowego poprzez modyfikację istniejącego już modelu w bibliotece NetLogo.

Konieczna była również analiza otrzymanych wyników i przedstawienie własnych refleksji.

1. Charakterystyka oryginalnego modelu

1) Wybrano model *Virus on a Network* z biblioteki *Networks* z Netlogo.

2) Cel modelu:

- Model ten wykorzystuje sieć powiązań między agentami, co można interpretować jako relacje między pracownikami.
- Model przedstawia rozprzestrzenianie się wirusa między agentami w sieci. Każdy agent może zostać *zarażony* przez swoich sąsiadów w sieci z określonym prawdopodobieństwem. Owego wirusa możemy potraktować jako *dezinformację/plotkę*.

3) Rodzaje agentów:

- Ludzie (*turtles*) - jednostki, które mogą się zarazić wirusem i rozprzestrzeniać go dalej
- Połączenia (*links*)- umożliwiają przepływ wirusa między ludźmi

4) Cechy agentów:

- Ludzie: mogą mieć stan
 - *infected* czyli agent jest zainfekowany i zaraża
 - *non-infected* czyli agent jest zdrowy (nie miał kontaktu z chorym)
 - *immune* czyli agent jest odporny i nie może być ponownie zarażony
- Połączenia nie mają dynamicznych stanów – są statyczne

5) Schemat działania agenta w modelu:

- a) Na początek symulacji każdy agent (osoba w sieci) zaczyna jako zdrowy i podatny na infekcję.
- b) Losowo wybrana grupa agentów zostaje zainfekowana – to tzw. *initial outbreak*. Zmieniają kolor na czerwony („zarażony i zaraża innych”).
- c) Krok symulacyjny:
 - Zainfekowani agenci próbują zarazić swoich sąsiadów (tych, z którymi mają połączenia w sieci).
 - Zarażenie zachodzi z określonym prawdopodobieństwem (*virus-spread-chance*).
- d) Co kilka kroków (gdy timer osiągnie wartość progową):
 - Agent sprawdza, czy może wyzdrowieć.
 - Jeśli tak:
 - Losowo zostaje odporny (nie może się więcej zarazić) – zmienia kolor na szary, albo

- Wraca do stanu podatnego (zdrowy, ale może znów się zarazić) – kolor niebieski.
- e) Symulacja kończy się, gdy nie ma już żadnych zarażonych agentów.

6) Dlaczego wybrano ten model: Jego struktura bardzo dobrze odwzorowuje sposób, w jaki informacje rozprzestrzeniają się w zespołach i organizacjach.

2. Charakterystyka NOWEGO modelu

1) Cel modelu:

- Nowy model ma odwzorować rozprzestrzeniania się plotki w zespole wraz z uwzględnieniem roli liderów.
- Model może badać wpływ mocy wpływu i liczby liderów na zasięg plotki w zespole i ilość osób która w nie uwierzy.

2) Zaplanowano wprowadzenie następujących zmian:

a) Zmiana kolorów agentów:

Członkowie zespołu:

- yellow – podatny (jeszcze nie słyszał plotki),
- red – uwierzył i szerzy plotkę,
- gray – zignorował plotkę i “uodpornił się”,

Połączenia: gray2 – kolor krawędzi od agentów ignorujących

b) Zmiana nazw zmiennych:

- infected -> uwierzył (czy uwierzył plotce)
- resistant -> ignoruje (czy jest uodporniony na plotkę)
- virus-check-timer -> plotka-timer (licznik do losowego testu)
- numer of nodes -> liczba-osób
- average-node-degree -> średnia-liczba-osób
- initial-outbreak-size -> początkowa-liczba-plotkarzy
- virus-check-frequency -> plotka-check-frequency
- virus-spread-chance -> prawd-rozpowiedzenia-plotki
- recovery-chance -> prawd-zatrzymania-plotki
- gain-resistance-chance -> prawd-uodpornienia-na-plotke

c) Zmiana wielkości zmiennych: zmieniono wielkości maksymalne liczby osób na 50 tak aby dotyczyła ona zespołu projektowego

3) Dodano nowego typu agentów:

- Wprowadzono nowy typ agentów– *lider*.
- Dzięki polu *typ* kod obsługuje różne reguły zachowań i kolorowania.
- Liderzy nie przenoszą plotki, nie są zależni od niej.
- Mają kolor *cyan* i są w kształcie *triangle*.
- Nie zmieniają miejsc podczas symulacji.
- Wpływają na prawdopodobieństwo uwierzenia w plotkę poprzez parametr *wplyw-liderow*

4) Dodano funkcjonalności:

- Rozszerzenie *spread-virus* – prawdopodobieństwo rozpowiedzenia plotki jest tłumione czynnikiem modyfikator zależnym od liczby liderów w sąsiedztwie.
- Nowy stan *czas-w-plotce* umożliwiający zmierzenie ile osoba z zespołu wierzy w plotkę
- *polacz-lidera k* – dla utworzonego lidera tworzy k krawędzi do najbliższych zwykłych agentów
- *dodaj-lidera* – tworzy nowego lidera na pozycji myszy i łączy go z czterema sąsiadami

5) Dodano elementy interfejsu:

- Button „dodaj lidera” – dodaje nowego lidera do planszy i łączy go 4 liniami z siecią
- Slider „liczba-liderów” – ustala ile liderów będzie w zespole min 0 max 5
- Slider „wpływ-lidera” (%) – wpływ lidera na prawdopodobieństwo uwierzenia w plotkę
- Plot „Średni czas wiary w plotkę” – wykres pokazujący jak w trakcie symulacji zmienia się średni czas uwierzenia w plotkę

6) Kod modelu

```
;; deklaracja agentów
turtles-own
[
  uwierzył?          ;; true, jeśli agent uwierzył plotkę
  ignoruje?         ;; true, jeśli agent ignoruje plotkę i jej nie
  powtarza
  plotka-timer      ;; ilość ticks od kiedy ostatnio agent był sprawdzany na
  posłuchanie się plotki
  typ               ;; "zwykły" lub "lider"
```

```

    czas-w-plotce      ;; ile ticków agent wierzy w plotkę
]

to setup
  clear-all ;; reste pola z poprzednich ustawień
  setup-nodes ;; tworzy agentów
  setup-spatially-clustered-network ;; tworzenie grupy zarażonych plotką
  ask n-of początkowa-liczba-plotkarzy turtles with [typ = "zwykly"] ;;
  losowanie grupy zarażonych na początku
    [ become-uwierzył ]
  ask links [ set color white ] ;; kolor linii połączeń
  reset-ticks

end

;; lider tworzy linki do K najbliższych zwykłych agentów
to polacz-lidera [ k ]
  ;; SELF to właśnie utworzony lider
  repeat k [
    let cel min-one-of turtles with [typ = "zwykly" and not link-neighbor?
myself] [distance myself]
    if cel != nobody [ create-link-with cel ]
  ]
end

to setup-nodes

  set-default-shape turtles "circle" ;; kształt agentów to kółko
  create-turtles liczba-osób ;; ilość agentów
  ;; --- siewca plotki (jeden) -----
  ;; --- siewca plotki (jeden) -----

  [
    set typ "zwykly"
    setxy (random-xcor * 0.9) (random-ycor * 0.9) ;; odleglosc od
lewej/prawej strony i gora/dół
    become-podatny ;; stan podatny na wpływ
    set plotka-timer random plotka-check-frequency
  ]
  create-turtles liczba-liderów [
    set typ "lider"
    set shape "triangle"
    set size 2
    set color cyan
    setxy (random-xcor * 0.9) (random-ycor * 0.9)
    set uwierzył? false
    set ignoruje? false
    set plotka-timer 0
  ]
end

to setup-spatially-clustered-network ;; sieć znajomości
  let num-links (średnia-liczba-osób * (liczba-osób + liczba-liderów)) / 2

```

```

while [count links < num-links ]
[
  ask one-of turtles
  [
    let choice (min-one-of (other turtles with [not link-neighbor?
myself])
                      [distance myself])
    if choice != nobody [ create-link-with choice ]
  ]
]
; make the network look a little prettier
repeat 10
[
  layout-spring turtles links 0.3 (world-width / (sqrt liczba-osób)) 1
]
end

to go
  if all? turtles [not uwierzył?] ;; symulacja konczy się kiedy wszyscy są
zdrowi
  [ stop ]
  ask turtles
  [
    set plotka-timer plotka-timer + 1
    if plotka-timer >= plotka-check-frequency
      [ set plotka-timer 0 ]
  ]
  ask turtles with [uwierzył?] [
    set czas-w-plotce czas-w-plotce + 1
  ]

  spread-virus
  do-virus-checks

  tick
end

to dodaj-lidera
  create-turtles 1 [
    set typ "lider"
    set shape "triangle"
    set size 2
    set color cyan
    set uwierzył? false
    set ignoruje? false
    set plotka-timer 0
    setxy mouse-xcor mouse-ycor
    polacz-lidera 4
  ]
end

to become-uwierzył
  set czas-w-plotce 0

  set uwierzył? true
  set ignoruje? false
  if typ = "zwykly" [ set color red ]
end

```

```

to become-podatny ;; turtle procedure
  set uwierzył? false
  set ignoruje? false
  if typ = "zwykly" [ set color yellow ]

end

to become-ignoruje ;; turtle procedure
  set uwierzył? false
  set ignoruje? true
  if typ = "zwykly" [ set color gray ]
  ask my-links [ set color gray - 2 ]
end

to spread-virus
  ask turtles with [uwierzył? and (typ = "zwykly" )]

  [
    ask link-neighbors with [typ = "zwykly" and not ignoruje? and not
uwierzył?]
    [
      let liderzy count link-neighbors with [typ = "lider"]      ;; ilu
liderów jest sąsiadami agenta

      let modyfikator max list 0 (1 - wpływ-lidera * liderzy)

      let skuteczna prawd-rozpowiedzenia-plotki * modyfikator

      if random-float 100 < skuteczna [
        become-uwierzył
      ]
    ]
  ]
end

to do-virus-checks
  ask turtles with [uwierzył? and plotka-timer = 0]
  [
    if random 100 < prawd-zatrzymania-plotki
    [
      ifelse random 100 < prawd-uodpornienia-na-plotke
      [ become-ignoruje ]
      [ become-podatny ]
    ]
  ]
end

```

Schemat nowego modelu:

1. Tworzenie agentów (setup-nodes):

- Tworzeni są zwykli agenci (żółci, kształt "circle") oraz liderzy (trójkąt, cyjanowi)
- Zwykli agenci są podatni na plotkę
- Liderzy są odporni na plotkę i pełnią funkcję stabilizującą (zmniejszając skuteczność przekazu)

2. Budowanie sieci (setup-spatially-clustered-network):

- Tworzenie połączeń między agentami na podstawie odległości (sieć społeczna)

3. Zainfekowanie startowych agentów (n-of początkowa-liczba-plotkarzy)

zenie gdy jest ich więcej niż 1.

Jeden lider trafiający w losowe miejsce w grafie może przypadkowo spowolnić lokalnie rozprzestrzenianie się plotki, odciągając uwagę modelu, ale nie tłumi głównego przepływu informacji.

4. Rozprzestrzenianie się plotki (spread-virus):

- Zwykli agenci, którzy wierzą w plotkę, próbują przekazać ją swoim sąsiadom
- Liderzy w otoczeniu zmniejszają skuteczność przekazania plotki (mechanizm wpływu)
- Przekaz następuje z prawdopodobieństwem prawd-rozpowiedzenia-plotki * modyfikator

5. Sprawdzenie, czy agenci przestają wierzyć w plotkę (do-virus-checks):

- Z określonym prawdopodobieństwem agenci porzucają plotkę
- Część z nich staje się odporna (ignoruje? = true), reszta wraca do stanu podatnego

5. Zakończenie symulacji:

- Gdy żaden agent już nie wierzy w plotkę (all? turtles [not uwierzył?])

3. Analiza działania modelu po przeprowadzeniu symulacji

Model został zaprojektowany jako symulacja rozprzestrzeniania się plotki w zespole, w którym występują różne typy agentów – przede wszystkim zwykli członkowie oraz liderzy.

Liderzy pełnią rolę stabilizującą, ograniczającą podatność otoczenia na przekaz plotki – ich obecność modyfikuje szansę, że dany agent ulegnie wpływowi innych.

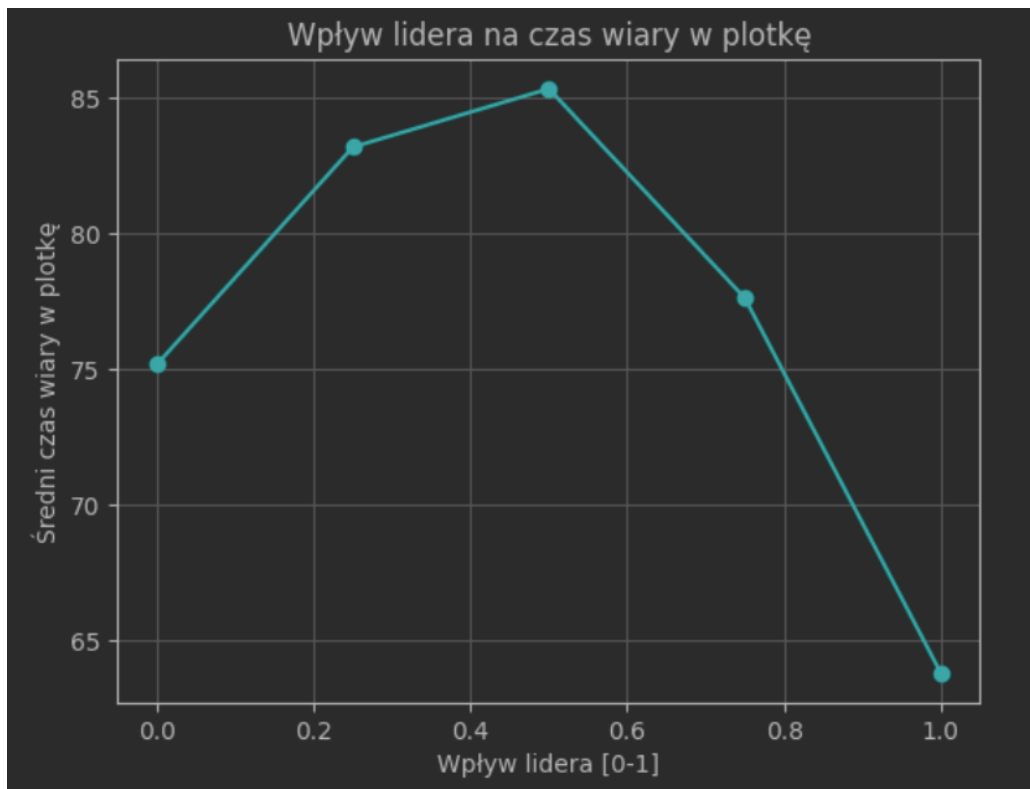
Wskaźnik "wpływ lidera" steruje siłą tego efektu, od 0 (liderzy nie mają wpływu) do 1 (pełna odporność przekazu w sąsiedztwie liderów).

W celu analizy wykonano następujące analizy:

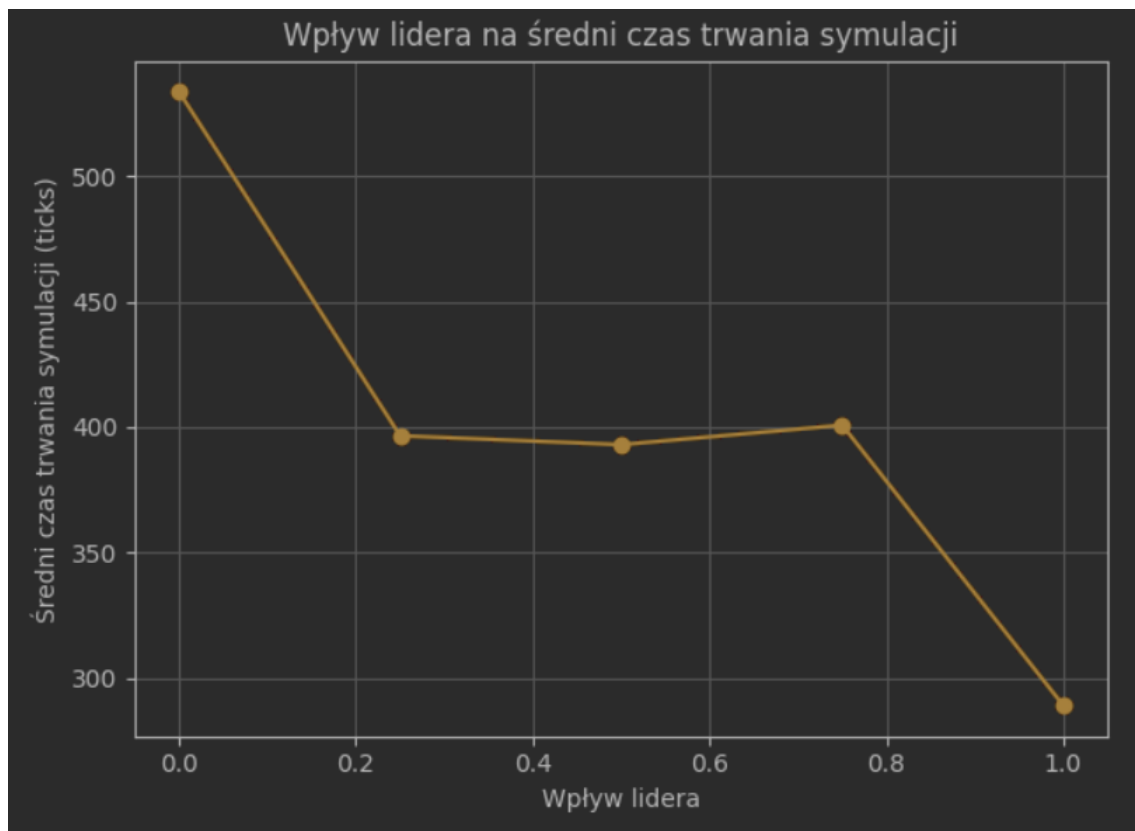
1. Wpływ lidera na liczbę osób, które uwierzyły w plotkę



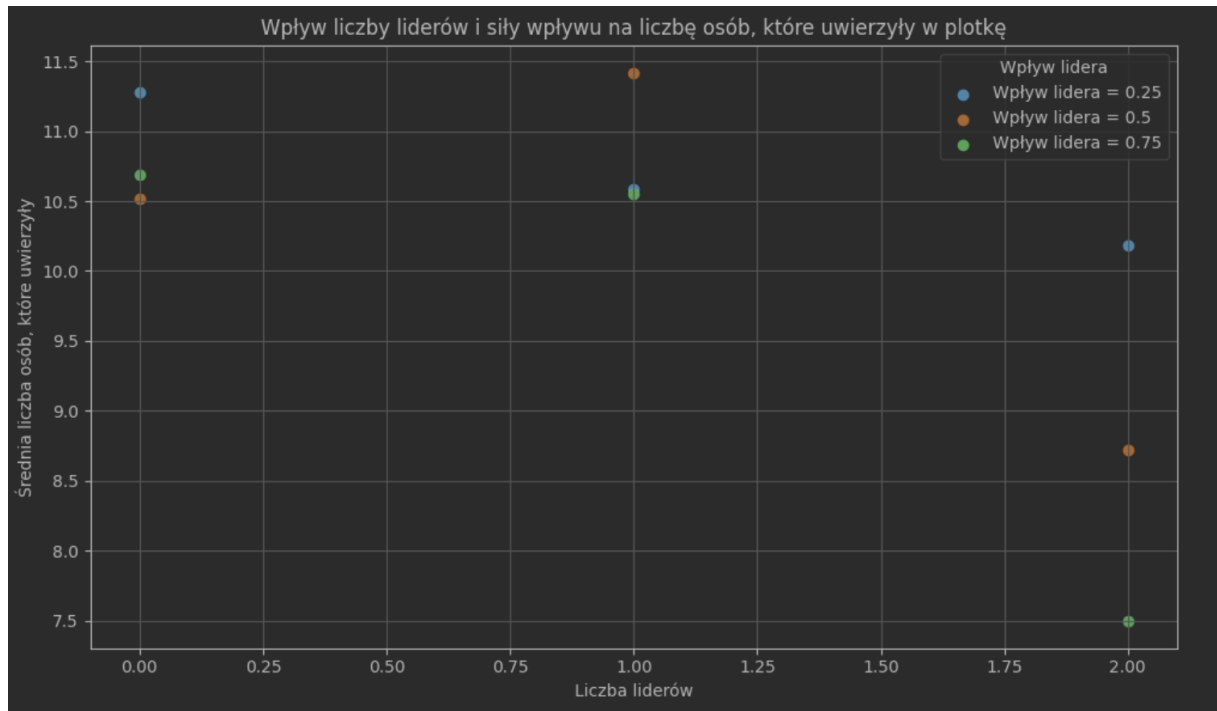
2. Wpływ lidera na czas wiary w plotkę



3. Wpływ lidera na średni czas trwania symulacji



4. Wpływ liczby liderów i siły wpływu na rozmiar zasięgu plotki



3. Wnioski końcowe

Na podstawie przeprowadzonych symulacji można zauważyć, że obecność liderów w zespole ma istotny wpływ na przebieg procesu rozprzestrzeniania się plotki. Zarówno liczba liderów, jak i poziom ich wpływu, odgrywają znaczącą rolę w ograniczaniu skali zjawiska.

Symulacje wykazały, że wraz ze wzrostem parametru wpływ lidera, liczba osób wierzących w plotkę systematycznie malała. W skrajnych przypadkach, gdy wpływ lidera był maksymalny, udawało się niemal całkowicie zahamować rozprzestrzenianie się informacji. Co ciekawe, największy średni czas wiary w plotkę występował przy średnich wartościach wpływu – liderzy spowalniali wtedy rozchodzenie.