

Modele segregacji przestrzennej i ich wykorzystanie w analizie urbanistycznej

Informatyka systemów złożonych

Aleksandra Szałajko
Wojciech Kordylewski-Kubus

semestr zimowy, 2022/2023

1 Wprowadzenie

Segregacja przestrzenna jest bardzo ważnym pojęciem odnoszącym się do narażenia danej klasy społecznej na zmiany środowiskowe, które w szerszym znaczeniu można rozumieć jako proces koncentracji różnych klas lub warstw społecznych w określonych regionach zamieszkania. Jest to przykład zagadnienia, którego cechy i uwarunkowania różnią się między krajami rozwiniętymi i rozwijającymi się: w krajach rozwiniętych segregacja przestrzenna jest często skorelowana z kwestiami etnicznymi i rasowymi, podczas gdy w krajach rozwijających się to różnice społeczno-ekonomiczne są podstawowym czynnikiem segregacji przestrzennej. Termin ten odnosi się do dowolnego dobrowolnego lub przymusowego oddzielenia obszarów mieszkalnych, między innymi ze względu na rasę, religię, pochodzenie etniczne, bądź też status ekonomiczny [2].

2 Cel i zakres projektu

Celem projektu jest implementacja algorytmów segregacji przestrzennej z uwzględnieniem cech populacji takich jak: rasa oraz status ekonomiczny. W projekcie zostaną zaimplementowane 4 algorytmy dokonujące segregacji ludności: algorytm Schellinga, algorytm Gaylorda D-Andria, dywergencja Kullback-Leiblera oraz dywergencja Bergmana. Zostaną przeprowadzone testy algorytmów z uwzględnieniem różnych cech populacji. Ostatecznie algorytmy te zostaną porównane względem poprawności oraz szybkości działania na wykorzystanych danych.

3 Podział aglomeracyjny

3.1 Opis algorytmu

Algorytm miał za zadanie wyznaczyć określoną ilość klastrow z dostarczonej macierzy. W pierwszym kroku każde pole traktowane jest jak osobna dzielnica. W kolejnych krokach klastry najbardziej podobne do siebie są łączone w coraz większe klastry, aż zostanie określona zadana ilość klastrow na planszy.

3.2 Dane

Macierz składa się z punktów, które oznaczają dzielnice. Każdy dystrykt posiada następujące atrybuty:

- procent ludności latynoskiej w dzielnicy
- procent białoskórych w dzielnicy
- procent czarnoskórych w dzielnicy
- procentowa ilość zarobków w odniesieniu do maksymalnej wypłaty dla ludności białej
- procentowa ilość zarobków w odniesieniu do maksymalnej wypłaty dla czarnoskórych
- procentowa ilość zarobków w odniesieniu do maksymalnej wypłaty dla Latynosów
- procentowa ilość mieszkańców posiadająca dzieci
- procentowa ilość mieszkańców posiadająca zwierzęta

3.3 Dywergencja

W celu obliczenia dywergencji między dwoma klastrami lub grupami używa się metod Bergmana lub Kullbacka-Leiblera.

1. Kullback-Leibler: $D_{KL}(P||Q) = \sum_i P(i) \log(P(i)/Q(i))$
2. Bergman: $D_B(P||Q) = \sum_i P(i) - Q(i)$

3.4 Implementacja algorytmu

Algorytm składa się z 3 klas:

3.4.1 Board

Klasa Board posiada następujące atrybuty: PERCENTAGE, ATRIBUTES, POPULATION.

Jej zadaniem jest stworzenie macierzy odzwierciedlającej miasto, dzielnice oraz wszelkie operacje na macierzy. Posiada następujące metody:

- procent ludności latynoskiej w dzielnicy
- generate board - funkcja tworząca macierz o podanej wielkości wypełnioną instancjami klasy District.

- count unique - funkcja zwracająca ilość grup i nie połączonych w grupę dzielnic występujących w macierzy.
- merge units - funkcja mająca za zadanie połączenie 2 dzielnic lub dołączenie dzielnicy do grupy dzielnic lub połączenie 2 grup.
- clusters raport - funkcja wyświetlająca informacje dotyczące grup i dystryktów. Informuje ona o średnich wartościach parametrów poszczególnych grup oraz wypisuje dystrykty wchodzące w skład grupy oraz wypisuje odchylenie standardowe wartości dla grup. Następnie wypisuje pojedyncze dystrykty, które nie zostały połączone w grupę.
- show board - funkcja wyświetlająca macierz, wyświetla połączone oraz samodzielne dystrykty.
- bergman divergence - funkcja statyczna przyjmująca parametry 2 dzielnic i wyliczająca dywergencję między nimi metodą Bergmana
- kullback leibler divergence - funkcja statyczna przyjmująca parametry 2 dzielnic i wyliczająca dywergencję między nimi metodą Kullbacka Liblera
- agglompartition - funkcja wyznaczająca dla podanej macierzy określoną ilość klastrów

```

1: function AGGLOMPARTITION( $R, k$ )
2: while  $|R| > k$  do
3:  $x_1^*, x_2^* \leftarrow \operatorname{argmin}_{(x_1, x_2) \in \mathcal{G}} d_{jb}(x_1, x_2)$ .
4: aggregate ( $x_1^*, x_2^*$ )
5: end while
6: return  $R$ 
7: end function

```

Rys. 1: Pseudokod funkcji podziału aglomeracyjnego [1]

3.4.2 District

Klasa typu 'dataclass' mająca za zadanie odzwierciedlenie pojedynczej dzielnicy. Poza wartościami opisanymi w sekcji 3.2 posiada ona pola: identifier - unikalny id dla dzielnicy, min_salary oraz max_salary określające odpowiednio minimalną oraz maksymalną możliwą średnią wypłatę. W post_init są losowane wartości atrybutów.

3.4.3 Group

Klasa dziedzicząca po klasie District mająca za zadanie odzwierciedlić połączone dzielnice. Zbiera

- add_unit - funkcja dodająca dystrykt do danej grupy
- update_group_info - funkcja obliczająca średnią charakterystykę dystryktów przynależących do grupy

- group_standard_deviation - funkcja wyliczająca odchylenie standardowe dla dzielnic przynależących do grupy.

3.5 Przykład działania algorytmu

Przykład algorytmu dla 9 dzielnic. Po zastosowaniu algorytmu i ustawieniu docelowej ilości klastrow na 5 otrzymujemy 2 grupy oraz 3 pojedyncze klastry.

```
[[18 19 20]
 [21 22 23]
 [24 25 26]]
Cluster:
Amount of units: 4, average cluster information: Group id: 27, black: 24%, white: 51%, latino: 21%, latino salary: 74%, black salary: 68%, white salary: 62%, pets: 44%, children: 35% population: 27380
Standard deviation: ('white': 14.236104336041748, 'latino': 13.524668018112232, 'black': 11.95477893619532, 'white_salary': 4.787135387016985, 'latino_salary': 23.15167380558045, 'black_salary': 15.577761302739723, 'children': 38.84155849945685, 'pets': 17.269916839178543)
District data: District id: 25, black: 15%, white: 44%, latino: 41%, latino salary: 94%, black salary: 58%, white salary: 68%, pets: 72%, children: 38% population: 5847
District data: District id: 23, black: 23%, white: 52%, latino: 25%, latino salary: 94%, black salary: 58%, white salary: 68%, pets: 37%, children: 16% population: 9578
District data: District id: 28, black: 42%, white: 48%, latino: 18%, latino salary: 52%, black salary: 88%, white salary: 57%, pets: 34%, children: 11% population: 6876
District data: District id: 26, black: 19%, white: 72%, latino: 9%, latino salary: 56%, black salary: 78%, white salary: 64%, pets: 46%, children: 9% population: 6599

Amount of units: 2, average cluster information: Group id: 28, black: 1%, white: 4%, latino: 94%, latino salary: 68%, black salary: 79%, white salary: 86%, pets: 69%, children: 43% population: 11988
Standard deviation: ('white': 12.028015208771307, 'latino': 12.727922861357855, 'black': 0.7871867811865476, 'white_salary': 7.0718678118654755, 'latino_salary': 0.0, 'black_salary': 6.363961838678928, 'children': 44.54772721475249, 'pets': 28.284271247461982)
District data: District id: 24, black: 1%, white: 8%, latino: 99%, latino salary: 68%, black salary: 77%, white salary: 89%, pets: 59%, children: 28% population: 8752
District data: District id: 22, black: 2%, white: 17%, latino: 81%, latino salary: 68%, black salary: 86%, white salary: 79%, pets: 99%, children: 91% population: 3156

Single districts:
District data: District id: 18, black: 18%, white: 82%, latino: 0%, latino salary: 94%, black salary: 85%, white salary: 88%, pets: 11%, children: 69% population: 7884
District data: District id: 19, black: 15%, white: 28%, latino: 57%, latino salary: 73%, black salary: 95%, white salary: 83%, pets: 24%, children: 73% population: 9719
District data: District id: 21, black: 74%, white: 23%, latino: 3%, latino salary: 91%, black salary: 74%, white salary: 97%, pets: 64%, children: 35% population: 6417
None
[[18 19 20]
 [21 20 27]
 [20 27 27]]
```

Rys. 2: Output z działania algorytmu

3.6 Eksperymenty

Na zaprojektowanym algorytmie przeprowadzono eksperyment polegający na porównaniu odchylenia standardowego cech populacji w klastrach po zastosowaniu dywergencji Kullbacka-Leiblera oraz dywergencji Bergmana. Badanie przeprowadzono na 25 dystryktach oraz docelowej ilości grup równej 10.

		Dywergencja	
		Bergman	Kullback-Leibler
Cechy populacji (wyrażone w procentach)	biali	11,54%	12,66%
	latynosi	13,33%	15,21%
	czarnoskórzy	11,45%	12,63%
	zarobki białych	11,77%	13,12%
	zarobki latynosów	11,76%	12,94%
	zarobki czarnoskórych	11,84%	13,67%
	domostwa posiadające potomstwo	18,95%	23,44%
	domostwa posiadające zwierzęta	18,66%	22,57%

Rys. 3: Porównanie odchylenia standardowego cech populacji dla stworzonych klastrow dla metod: Bergmana oraz Kullbacka-Leiblera

4 Modelowanie oparte na agentach

Modelowanie oparte na agentach to generatywne podejście do badania zjawisk naturalnych oparte na interakcji jednostek w systemach społecznych, fizycznych lub biologicznych.

4.1 Dane

Dane zostały wygenerowane losowo jako 4 odrębne populacje:

1. ludność czarnoskóra biedna
2. ludność czarnoskóra bogata
3. ludność białoskóra biedna
4. ludność białoskóra bogata

Jako przestrzeń działania algorytmu zastosowano macierz o rozmiarze 80x80 z periodycznymi warunkami brzegowymi. Agenci są losowo rozmieszczani na planszy. Prawdopodobieństwo wystąpienia poszczególnych populacji jest sobie równe, podczas gdy gęstość pustych przestrzeni jest modyfikowalnym parametrem algorytmu.

4.2 Opis algorytmów

W projekcie zaimplementowano model Schellinga oraz Gaylorda D'Andria. Oba algorytmy są agentowymi modelami segregacji przestrzennej. Opierają się na analizie najbliższego sąsiedztwa i decyzji czy agent jest nim usatysfakcjonowany pod względem podobieństwa. Model Schellinga umożliwia migrację agentów jedynie w obszarze własnego sąsiedztwa, natomiast algorytm Gaylorda D'Andria pozwala na dowolnie odległe migracje. Oba modele prowadzą do powstawania dużych skupisk agentów tego samego typu [3].

Wśród parametrów modeli można wyróżnić:

- *agents_ratio* - procent miejsc zajętych przez agentów
- α - liczba agentów tej samej rasy oraz o tym samym statusie materialnym, która jest wystarczająca, aby agent nie chciał zmienić miejsca zamieszkania
- *α_{rase}* - liczba agentów tej samej rasy, która jest wystarczająca, aby agent nie chciał zmienić miejsca zamieszkania
- *α_{salary}* - liczba agentów o tym samym statusie materialnym, która jest wystarczająca, aby agent nie chciał zmienić miejsca zamieszkania
- *n_steps* - ilość iteracji algorytmu

4.3 Implementacja algorytmów

4.3.1 Model Schellinga

1. Wygeneruj i losowo umieść agentów na planszy
2. Dla wybranej liczby iteracji:

- (a) Dla każdego agenta:
 - i. Sprawdź najbliższe sąsiedztwo
 - ii. Wyznacz liczbę podobnych sąsiadujących agentów względem rasy i statusu materialnego, jedynie rasy, jedynie statusu materialnego
 - iii. Jeśli ilość podobnych sąsiadów jest większa lub równa odpowiednio α , α_{rase} , α_{salary} to pozostań w obecnym miejscu
 - iv. W przeciwnym wypadku:
 - A. Sprawdź czy w sąsiedztwie są wolne miejsca
 - B. Jeśli nie to pozostań w obecnym miejscu
 - C. Jeśli tak to wybierz dowolne z nich jako możliwe miejsce migracji
- (b) Sprawdź wszystkie wolne miejsca, które zostały wybrane przez agentów jako możliwe miejsce migracji i jeśli dwóch lub więcej agentów wybrało to samo wolne miejsce do przeprowadzki, to zrezygnuj ze zmiany miejsca
- (c) Przenieś agentów, którzy zdecydowali się na zmianę lokalizacji

4.3.2 Model Gaylorda D'Andria

1. Wygeneruj i losowo umieść agentów na planszy
2. Wykonuj do momentu, w którym każdy agent będzie usatysfakcjonowany swoim miejscem zamieszkania:
 - (a) Dla każdego agenta:
 - i. Sprawdź najbliższe sąsiedztwo
 - ii. Wyznacz liczbę podobnych sąsiadujących agentów względem rasy i statusu materialnego, jedynie rasy, jedynie statusu materialnego
 - iii. Jeśli ilość podobnych sąsiadów jest większa lub równa odpowiednio α , α_{rase} , α_{salary} to pozostań w obecnym miejscu
 - iv. W przeciwnym wypadku:
 - A. Dodaj agenta do listy agentów chętnych do zmiany lokalizacji
 - (b) W przypadku pustego miejsca:
 - i. Dla każdej z populacji:
 - A. Sprawdź czy puste miejsce spełnia warunek odpowiedniej liczby podobnych sąsiadujących agentów względem rasy i statusu materialnego, jedynie rasy, jedynie statusu materialnego
 - B. Jeśli tak to dodaj je do listy miejsc odpowiednich do przeprowadzki
 - (c) Sprawdź wszystkie wolne miejsca odpowiednie do przeprowadzki i jeśli dwie lub więcej populacji wybrało to samo wolne miejsce do przeprowadzki, to usuń je z listy miejsc odpowiednich do przeprowadzki
 - (d) Jeśli długości list obejmujących agentów chętnych do zmiany oraz miejsca odpowiednie do przeprowadzki dla każdej populacji są różne, skróć dłuższą z nich usuwając losowo elementy
 - (e) Przenieś agentów z każdej populacji na losowe miejsce z listy miejsc odpowiednich do przeprowadzki

4.4 Przeprowadzone eksperymenty

Wykonano po 3 symulacje dla każdego z zaimplementowanych algorytmów. Celem było porównanie wpływu ilości miejsc zajętych przez agentów na wyniki segregacji przestrzennej wewnątrz każdego z algorytmów oraz porównanie ze sobą działania obu algorytmów.

W modelu Schellinga przyjęto następujące parametry:

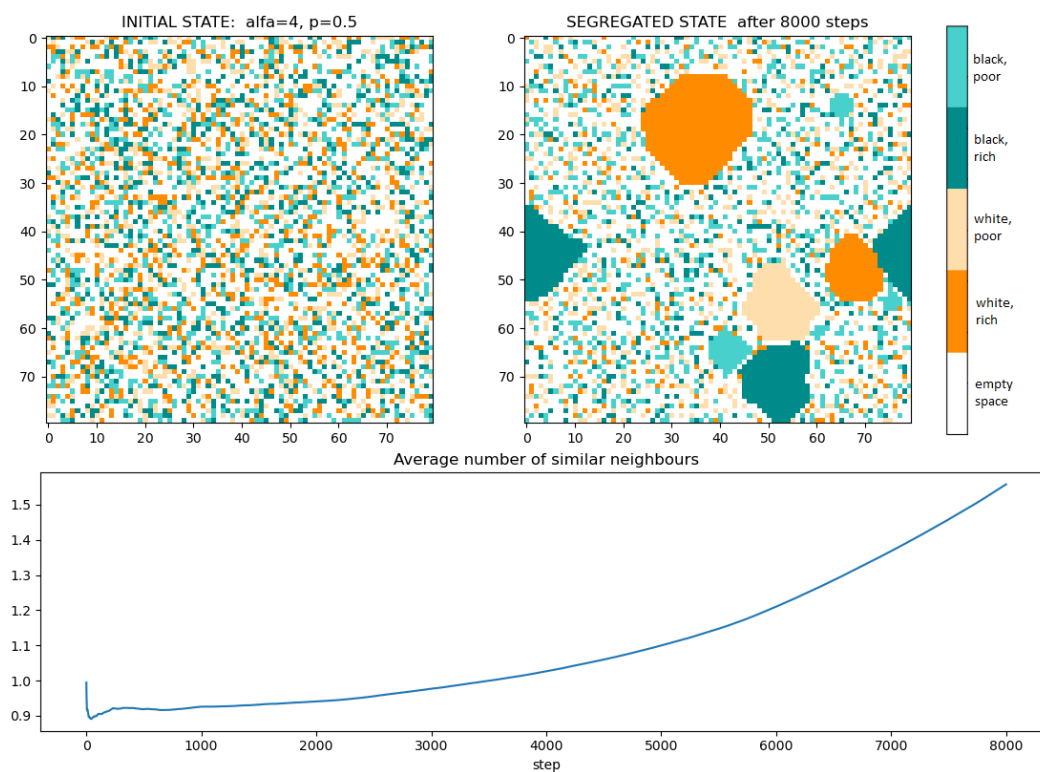
- $\alpha = 4$
- $\alpha_{rase} = 6$
- $\alpha_{salary} = 6$
- $n_{steps} = 8000$

W modelu Gaylorda D’Andria przyjęto następujące parametry:

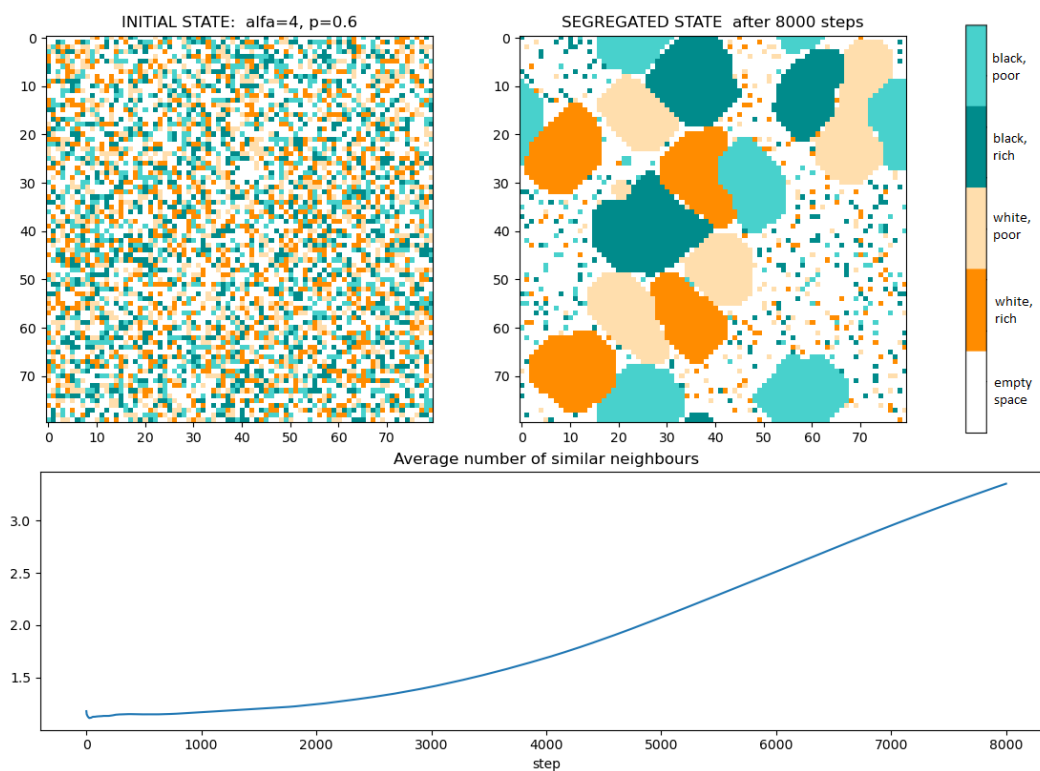
- $\alpha = 3$
- $\alpha_{rase} = 6$
- $\alpha_{salary} = 6$

Parametry α , α_{rase} , α_{salary} zostały ustalone poprzez przetestowanie różnych wartości i uwzględnienie tych, które dają najlepsze wyniki segregacji dla danego modelu. Dla obu algorytmów wśród parametru $agents_ratio$ przetestowano wartości: 0.5, 0.6, 0.7.

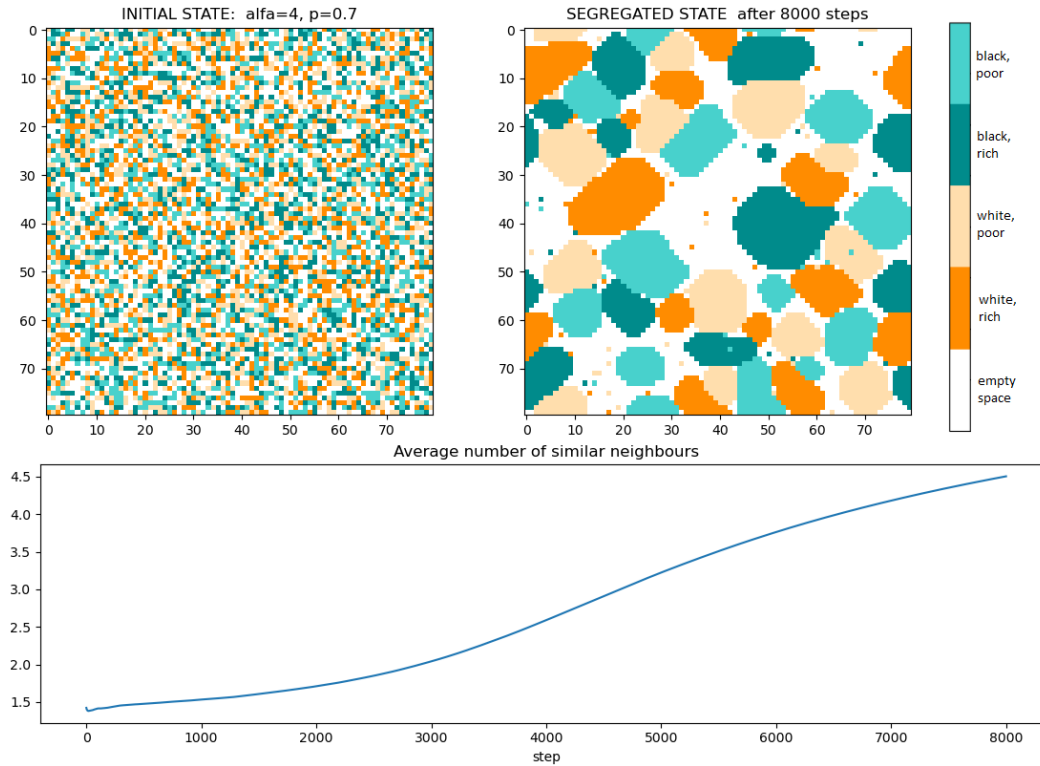
Następnie przeprowadzono odpowiednie symulacje wraz z wyrysowaniem wykresu przedstawiającego zmianę średniej ilości podobnych sąsiadów w trakcie działania algorytmu.



Rys. 4: Przeprowadzona symulacja modelu Schellinga na planszy w 50% wypełnionej przez agentów



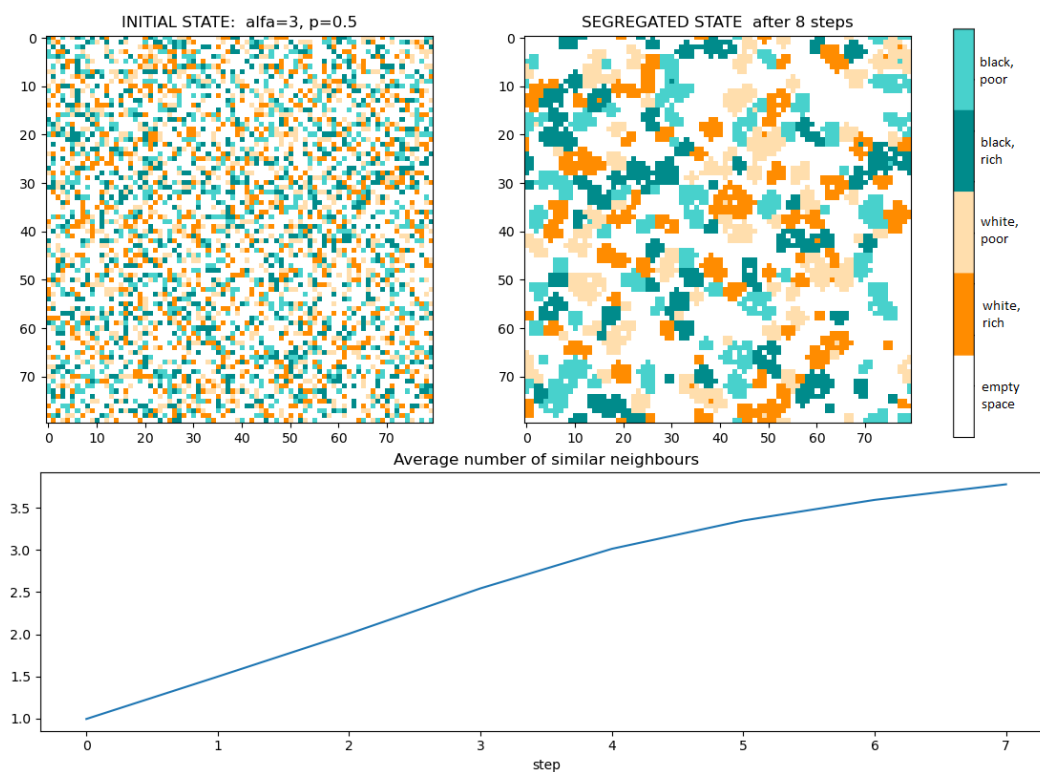
Rys. 5: Przeprowadzona symulacja modelu Schellinga na planszy w 60% wypełnionej przez agentów



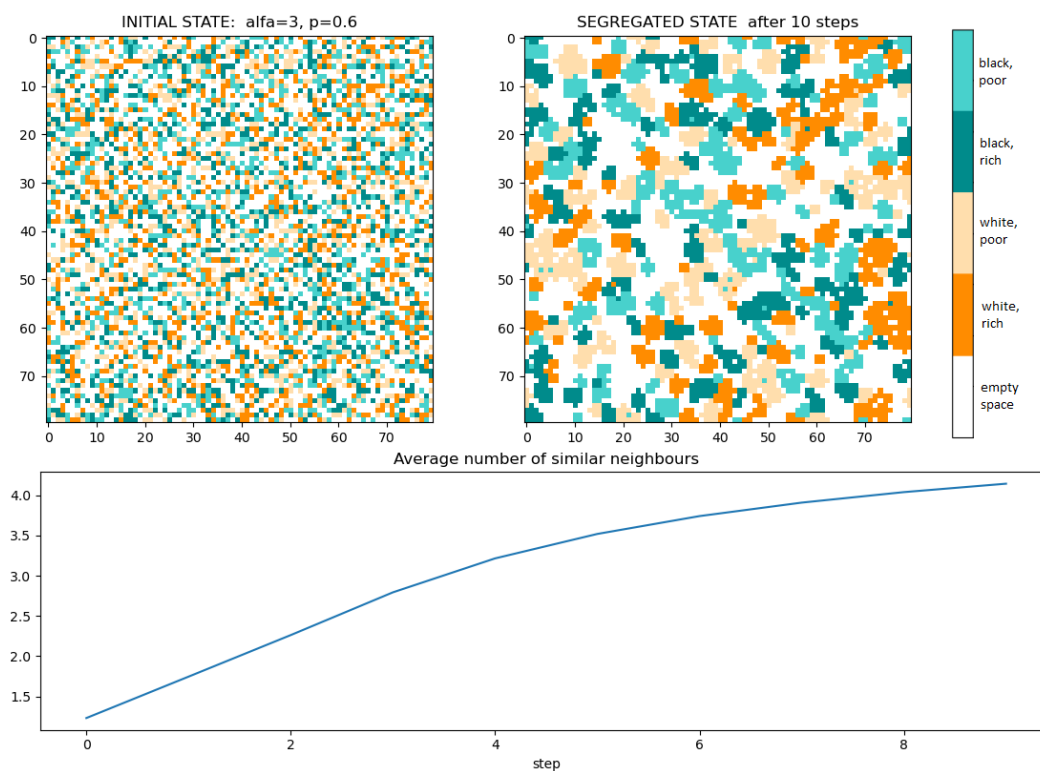
Rys. 6: Przeprowadzona symulacja modelu Schellinga na planszy w 70% wypełnionej przez agentów

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że model Schellinga jest w stanie dokonać segregacji przestrzennej. Im większy procent wypełnienia planszy przez agentów, tym szybciej algorytm dokonuje segregacji populacji. Można zauważyć, że wartość procentu wypełnienia planszy ma również wpływ na kształt oraz ilość powstałych w wyniku działania algorytmu klastrów. W przypadku planszy w 50% wypełnionej przez agentów klastrów jest niewiele, ale są duże. Natomiast im więcej agentów, tym klastry stają się mniejsze i jest ich znacznie więcej.

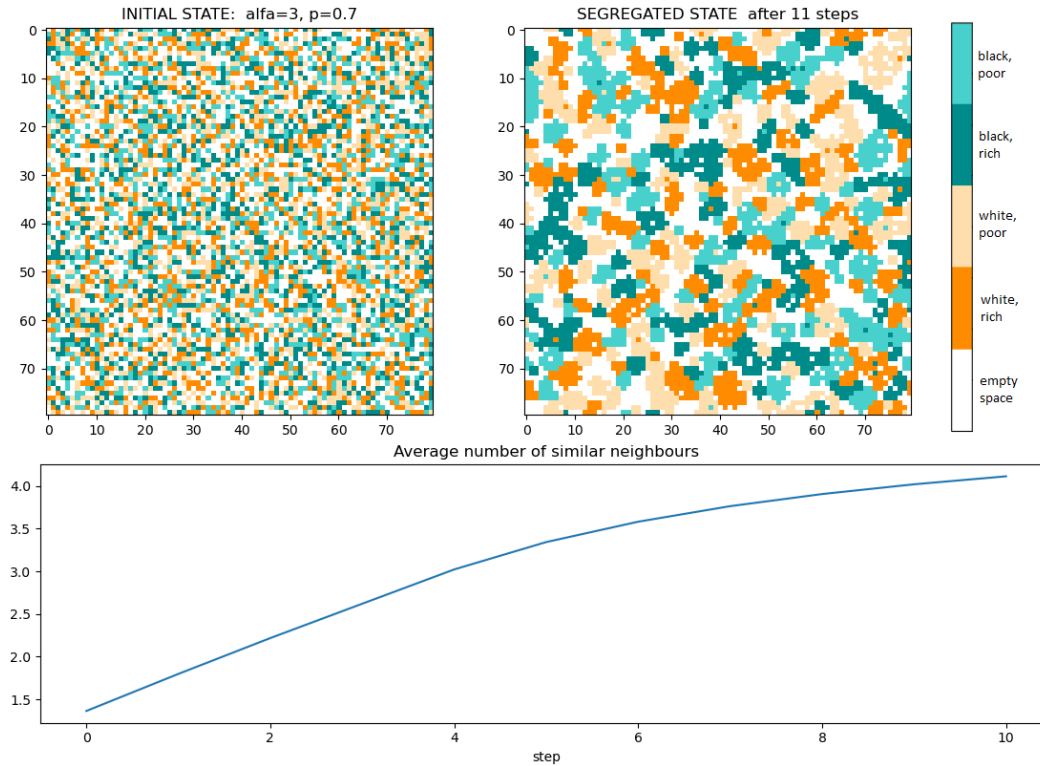
Na podstawie wykresów przedstawiających zmianę średniej ilości podobnych sąsiadów w trakcie działania algorytmu, można stwierdzić, że im więcej wolnych przestrzeni tym wartość ta jest niższa przez cały okres działania algorytmu. W przypadku planszy w 50% wypełnionej przez agentów na początku symulacji wartość średniej ilości podobnych sąsiadów drastycznie spada, po czym stopniowo wzrasta, jednak bardzo powoli. Dopiero po około 3500 iteracjach wraca do wartości początkowej. Po 8000 iteracjach osiąga wartość jedynie 1.55, co nie wskazuje na całkowitą segregację przestrzenną. Aby osiągnąć całkowitą segregację przestrzenną w przypadku planszy w 50% wypełnionej przez agentów konieczne jest wykonanie większej ilości iteracji algorytmu. W przypadku pozostałych testowanych parametrów: plansza w 60% oraz 70% wypełniona przez agentów, wykres nie cechuje się już tak drastycznym spadkiem wartości średniej ilości podobnych sąsiadów na początku symulacji oraz wzrasta znacznie szybciej podczas działania algorytmu. W przypadku wypełnienia w 70% dochodzi do całkowitej segregacji przestrzennej, na co wskazuje wartość średniej ilości podobnych sąsiadów, która pod koniec działania algorytmu jest równa 4.5, a więc przekracza parametr α .



Rys. 7: Przeprowadzona symulacja modelu Gaylorda D'Andria na planszy wypełnionej w 50%



Rys. 8: Przeprowadzona symulacja modelu Gaylorda D'Andria na planszy wypełnionej w 60%



Rys. 9: Przeprowadzona symulacja modelu Gaylorda D'Andria na planszy wypełnionej w 70%

Na podstawie przeprowadzonych symulacji algorytmu Gaylorda D'Andria widać, iż jego działanie prowadzi do segregacji przestrzennej. W przeciwieństwie do algorytmu Schellinga, im większy procent wypełnienia planszy przez agentów, tym wolniej algorytm dokonuje segregacji populacji. Jednak jest to jedynie różnica 1-3 iteracji algorytmu pomiędzy przeprowadzonymi symulacjami. W przypadku planszy w 50% wypełnionej przez agentów klastry przyjmują kształt zbliżony do koła, podczas gdy zwiększenie tego parametru prowadzi do powstawania klastrów o nieforemnych kształtach.

W przypadku wszystkich symulacji wartość średniej ilości podobnych sąsiadów wzrasta już od pierwszej iteracji algorytmu i wystarczy jedynie 8-11 iteracji aby osiągnąć segregację przestrzenną. We wszystkich 3 zbadanych przypadkach wartość średniej ilości podobnych sąsiadów przekracza parametr α .

Wewnątrz niektórych klastrów widać pojedyncze punkty o innym kolorze. Symbolizują one agentów, którzy zdecydowali się pozostać wśród sąsiedztwa, z którym są połączeni jedynie jedną z cech: rasą, bądź statusem materialnym.

5 Podsumowanie i wnioski

Algorytm podziału aglomeracyjnego miał za zadanie wyznaczenie określonej ilości klastrow z dzielnicami o jak najmniejszej dywergencji na dostarczonej planszy. W tym celu zaimplementowano 3 klasy, które odzwierciedlały plansze, dystrykt oraz grupę dzielnic. Na zaimplementowanym algorytmie sprawdzono 2 metody dywergencji: Bergmana i Kullbacka-Leiblera. Zauważono, że algorytm, który wykorzystywał dywergencję Bergmana posiadał mniejsze odchylenie standardowe między dzielnicami w grupach. Ponad to algorytm dla małej docelowej ilości grup tworzył jedną dużą grupę oraz pojedyncze dystrykty.

Z przeprowadzonych eksperymentów wynika, że zarówno algorytm Schellinga, jak i Gaylorda D'Andria są w stanie dokonać segregacji przestrzennej wśród 4 populacji. Model Schellinga prowadzi do powstania znacznie większych i kulistych klastrow, podczas gdy model Gaylorda D'Andria tworzy więcej, ale znacznie mniejszych klastrow o wydłużonym, nieforemnym kształcie. Algorytm Schellinga wymaga bardzo dużej ilości iteracji, aby dokonać segregacji (nawet ponad 8 tysięcy w przypadku 50% zapelnienia planszy przez agentów), natomiast w przypadku modelu Gaylorda D'Andria wystarczy ich jedynie 8-11. Dlatego też wydajność algorytmów znacznie się różni - symulacja algorytmu Gaylorda D'Andria trwa kilkanaście sekund, z kolei model Schellinga wymaga kilku godzin, by dokonać segregacji przestrzennej na wykorzystanych danych.

Link do repozytorium: <https://github.com/wojtekkk99/Spatial-segregation>

Bibliografia

- [1] Philip S. Chodrow. „Structure and information in spatial segregation”. W: *Proceedings of the National Academy of Sciences* 114.44 (2017), s. 11591–11596. DOI: 10.1073/pnas.1708201114. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1708201114>.
- [2] Marcos Roberto Martines i in. „Spatial segregation in floodplain: An approach to correlate physical and human dimensions for urban planning”. W: *Cities* 97 (2020), s. 102551. ISSN: 0264-2751. DOI: <https://doi.org/10.1016/j.cities.2019.102551>.
- [3] Diego Ortega, Javier Rodríguez-Laguna i Elka Korutcheva. „Segregation in spatially structured cities”. W: *Physica A: Statistical Mechanics and its Applications* 608 (2022), s. 128267. ISSN: 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2022.128267>.