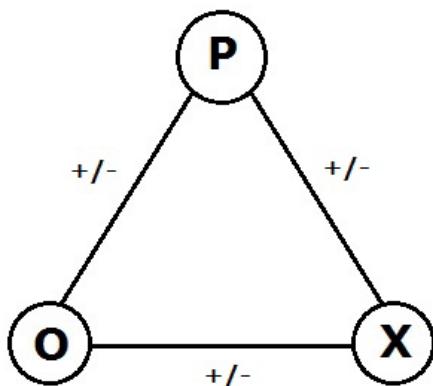


3. Model Heidera

Opis i reguły modelu

Model Heidera jest jedną z implementacji wywodzącej się z psychologii teorii równowagi, którą zaproponował właśnie sam Fritz Heider. Teoria bada zmiany w nastawieniu m.in. członków populacji, gdzie każdy z nich dąży do spójności poznawczej – potrzeby podtrzymywania własnych wartości i opinii – w celu uzyskania równowagi psychicznej.

Między członkami (ale także bytami, przedmiotami) mogą zaistnieć relacje pozytywne, jak i negatywne (np. lubi, nie lubi). Heider w swoim modelu P-O-X zaproponował triady (składające się z członków P, O oraz X) z właśnie takimi relacjami.



Balans poznawczy ma miejsce, kiedy w triadzie występują dwie relacje negatywne i jedna pozytywna lub wszystkie są pozytywne:

$$- - + = +$$

$$+ + + = +$$

Może także dojść do dysonansu poznawczego, który w psychologii opisywany jest jako mentalny niepokój, który wynika z niespójnych lub sprzecznych, np.: zachowań czy przekonań w porównaniu do aktualnie znanych i posiadanych, co prowadzi do potrzeby zmiany w celu zmniejszenia zaistniałego dysonansu. Problemy mogą zaistnieć m.in. w obszarze decyzji, uczuć, opinii, przekonań czy samego środowiska. W sieci sytuacja ta objawia się jako dwie relacje pozytywne i jedna negatywna lub wszystkie negatywne:

$$+ + - = -$$

$$- - - = -$$

W triadach zbalansowanych znaki relacji odpowiadają zasadom:

- przyjaciel mojego przyjaciela jest moim przyjacielem,
- przyjaciel mojego wroga jest moim wrogiem,
- wróg mojego przyjaciela jest moim wrogiem,
- wróg mojego wroga jest moim przyjacielem.

Symulacje opisane w sprawozdaniu zostały przeprowadzone na własnej implementacji modelu na grafie zupełnym z liczbą wierzchołków (aktorów) $N = 50$, bez uwzględnienia szumu.

Zaimplementowano zarówno synchroniczną, jak i asynchroniczną wersję modelu. Realizowany model Heidera opiera się na kilku regułach:

1. Między każdą parą węzłów (aktorów) i, j w sieci przypisana zostaje relacja x_{ij} o wartości $x_{ij} = 1$ – relacja sympatii lub $x_{ij} = -1$ – relacja wrogości.
2. Triady relacji, które nie są zbalansowane, należy zbalansować. W takim przypadku każdą relację (krawędź) należy sprowadzić do wartości znaku, który otrzymamy przez zsumowanie iloczynów wszystkich innych sąsiadujących par krawędzi tworzących triady z aktualną krawędzią.

$$x_{ij}(t+1) = \text{sign} \left(\sum_n x_{in}(t) x_{nj}(t) \right)$$

n – sąsiad pary i, j

3. Zbalansowanie całego układu U badamy przez zsumowanie iloczynów relacji (krawędzi) wszystkich możliwych triad do utworzenia w sieci, znormalizowanych przez liczbę tych triad, pomnożoną przez -1 . $U = -1$ oznacza sieć zbalansowaną.

$$U = -\frac{1}{\Delta} \sum_i \sum_{j>i} \sum_{k>j} x_{ij} x_{jk} x_{ik}$$

Każda sieć po dostatecznie długim czasie kończy działanie z ustaleniem relacji wszystkich członków na zaprzyjaźnionych całkowicie ($U = -1$, $\langle x_{ij} \rangle = 1$) lub zaprzyjaźnionych we własnych grupach (dwóch), ale wrogich między sobą ($U = -1$, $\langle x_{ij} \rangle \neq 1$).

Wynika to z faktu, iż dla U iloczyny $1 \times 1 \times 1$ oraz $-1 \times -1 \times 1$ (które są „bokami” poszczególnych triad) dają wartość 1, więc po podzieleniu ich sumy przez liczbę triad i pomnożeniu przez -1 otrzymujemy -1 . Przy $\langle x_{ij} \rangle$ również, gdy wszystkie wierzchołki są dodatnie i podzielimy ich wartości (1) przez liczbę, otrzymamy 1, natomiast przy występowaniu wartości -1 (przy triadach $-1 \times -1 \times 1$) otrzymamy wynik poniżej wartości 1.

Implementacja sieci do symulacji modelu bazuje na języku programowania Python oraz bibliotece *networkx*. Biblioteka pozwoliła stworzyć czytelne rozwiążanie dla relacji w sieci z zastosowaniem grafu zupełnego. Wizualizacje grafów także powstały przez połączenie tej biblioteki oraz biblioteki *matplotlib*.

Zadanie

Opis

Utworzona została sieć w oparciu o graf zupełny z liczbą wierzchołków $N = 50$. Każda relacja (krawędź) przyjmuje losowo wartość 1 lub -1 z prawdopodobieństwem 50:50.

Eksperyment miał na celu prześledzenie zmian relacji jako całości w układzie od chwili początkowej $t = 0$ oraz stan końcowy efektów balansowania. Po każdej całkowitej aktualizacji sieci zapamiętywane i wypisywane zostają wartości: chwila czasowa t , wartość U oraz wartość $\langle x_{ij} \rangle$. Kiedy U oraz $\langle x_{ij} \rangle$ nie ulegają już aktualizacji, wypisane zostają wszystkie pary relacji i, j wraz z ich wartością x_{ij} , a także ostateczny wygląd sieci.

W wersji z aktualizacją synchroniczną początkowo zapamiętywane są wartości balansowania triad dla każdej relacji z osobna, a dopiero na samym końcu wszystkie zostają zaktualizowane do nowych wartości.

W wersji z aktualizacją asynchroniczną aktualizacja każdej kolejnej relacji (krawędzi) opiera się na już wcześniej zaktualizowanych wartościach innych relacji, które są wynikiem balansowania triad dla poprzednich relacji.

Wynikiem jest sieć, w której triady relacji są zbalansowane w taki sposób, jak opisano w regułach modelu opisanych wcześniej w sprawozdaniu.

W załączonym pliku źródłowym do zadania *main.py* zostało opisane działanie kodu.

W tym samym pliku zaimplementowano zarówno wersję synchroniczną, jak i asynchroniczną dla modelu (*update_edges_weight_sync* i *update_edges_weight_async*). Funkcje wywołujące wersje znajdują się obok siebie, jedną z nich należy odkomentować/ zakomentować. Również funkcja (*create_complete_graph*) tworząca graf z prawdopodobieństwem podziału wartości relacji 1 i -1 50:50 jest domyślnie odkomentowana, sugerowany rozkład 65:35 należy odkomentować – propozycja takiego rozkładu wynika ze znacznie łatwiejszego (częstszego) uzyskania stanu raju. Taki rozkład ustalono po odbyciu wielu prób symulacji. Miejsce opisywanych par funkcji oznaczono komentarzem „-- !! --”.

Wyniki symulacji

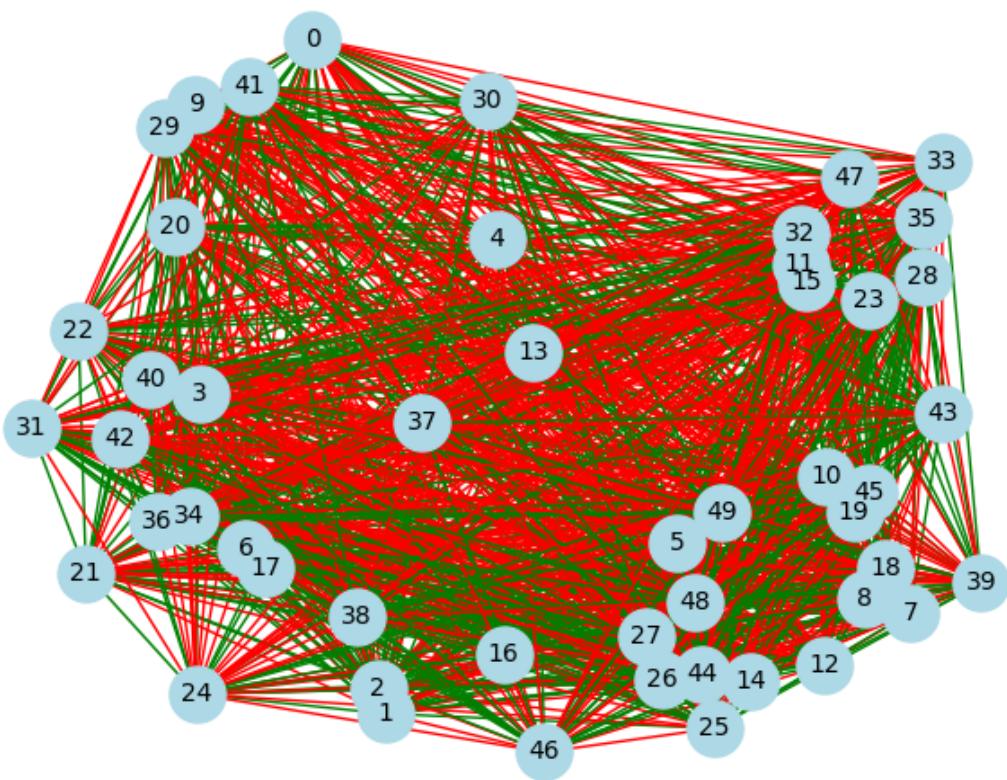
Prześledzone zostały zmiany dla sieci w wersji synchronicznej i asynchronicznej. Na grafikach przedstawiono sieć w chwili czasowej $t = 0$ oraz końcowy wygląd sieci dla stanu podziału na grupy i raju (tutaj udało się uzyskać przy podziale 65:35). Nie dzielono grafik na wersję synchroniczną i asynchroniczną, ponieważ nie ma między nimi widocznej różnicy. W załączonych plikach w dokumencie tekstowym można zapoznać się z dokładniejszymi wynikami symulacji.

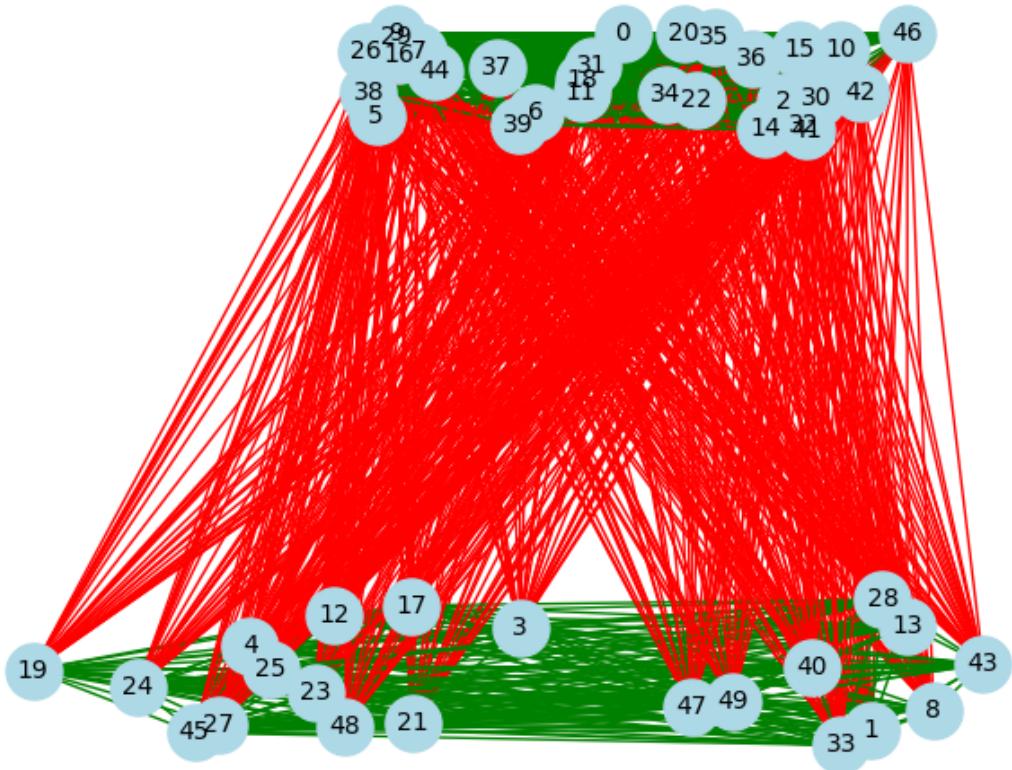
W wersji synchronicznej do stanu zbalansowania sieć potrzebuje tych kroków więcej, jest ich najczęściej w okolicach 6. Jednak po obserwacji wielu symulacji końcowa liczba relacji pozytywnych (1) często wydaje się większa, co można zobaczyć po wartości końcowej $\langle x_{ij} \rangle$.

W wersji asynchronicznej można zauważyc, że sieć potrzebuje mniejszej liczby kroków, aby dojść do zbalansowania. Zazwyczaj są to 3-4 kroki.

Dla prawdopodobieństwa 50:50 w obu wersjach bardzo często dochodzi do stanu podziału na dwie grupy zaprzyjaźnione, ale wrogie między sobą. Stan raju zdarza się bardzo rzadko, w odbytych symulacjach się nie zdarzył. W związku z tym, po wielu odbytych próbach symulacji można zauważyc, że w okolicach rozkładu 65:35 stan ten zaczyna pojawiać się częściej i w mniejszej liczbie iteracji dla wersji synchronicznej (średnio 4 kroki) i z niewielką poprawą dla wersji asynchronicznej (zazwyczaj 2-3 kroki).

Podział na grupy – wzajemnie wrogie, wewnętrznie zaprzyjaźnione:





t, U, avg_xij: 0, 0.00040816326530612246, 0.030204081632653063

t, U, avg_xij: 1, -0.07326530612244898, 0.03183673469387755

t, U, avg_xij: 2, -0.14265306122448979, 0.008979591836734694

t, U, avg_xij: 3, -0.29244897959183674, 0.036734693877551024

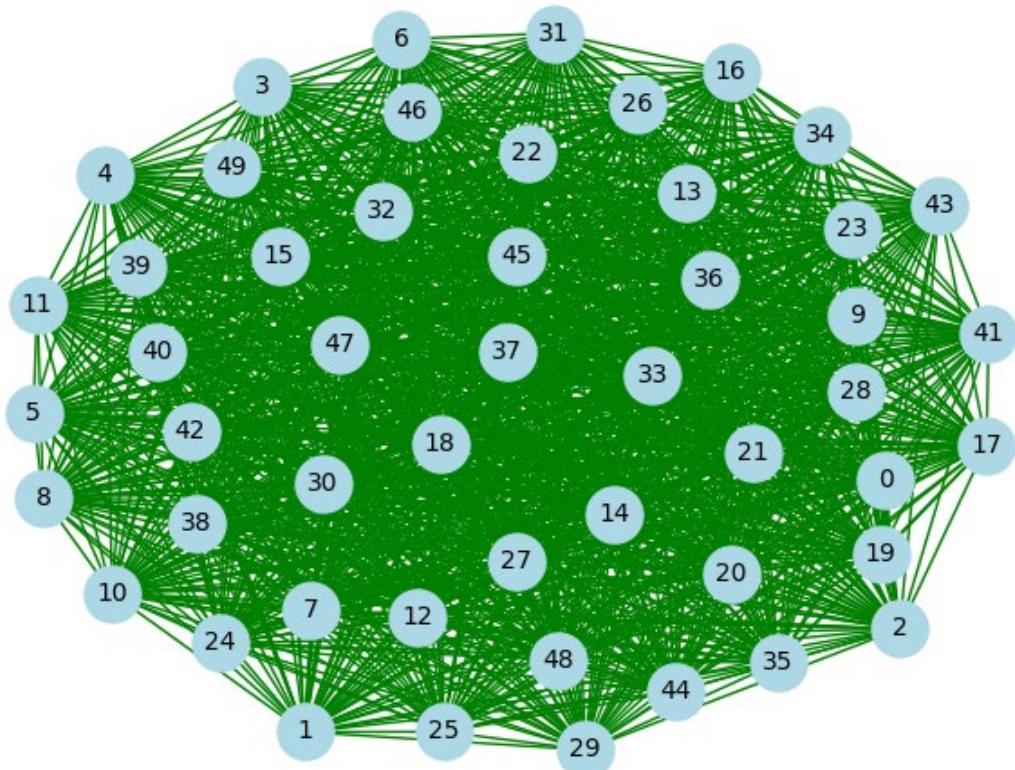
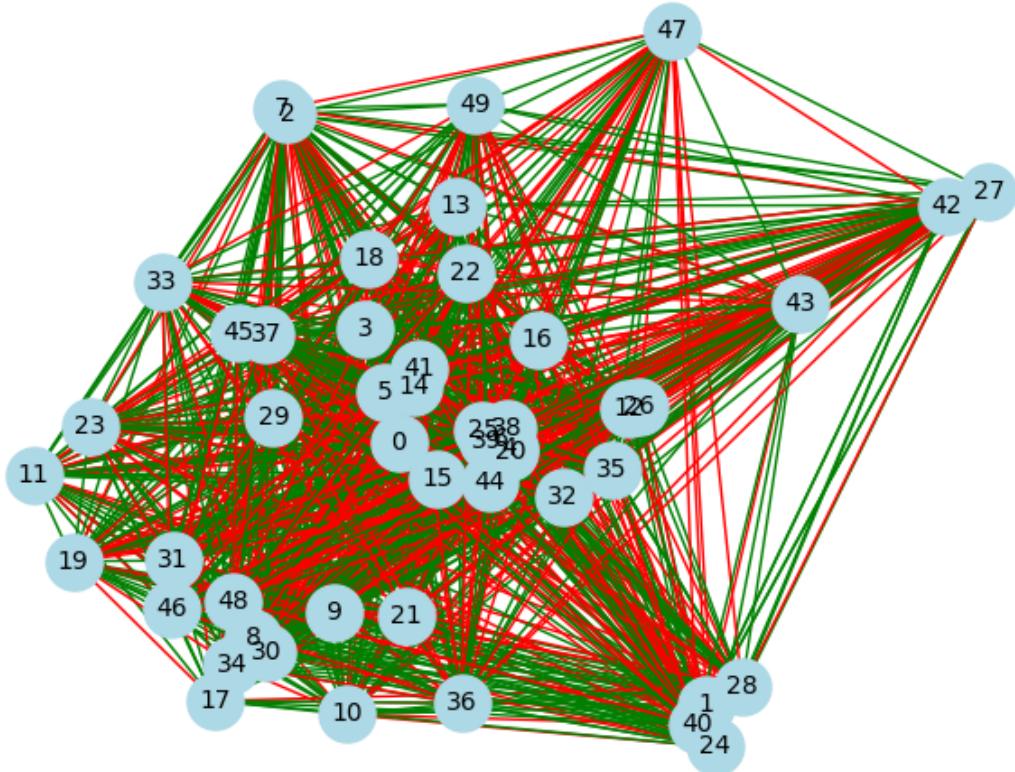
t, U, avg_xij: 4, -0.49918367346938775, 0.004081632653061225

t, U, avg_xij: 5, -0.7820408163265307, 0.0024489795918367346

t, U, avg_xij: 6, -0.9951020408163266, 0.007346938775510204

t, U, avg_xij: 7, -1.0, 0.005714285714285714

Stan raju:



t, U, avg_xij: 0, -0.0024489795918367346, 0.2653061224489796

t, U, avg_xij: 1, -0.12081632653061225, 0.3763265306122449

t, U, avg_xij: 2, -0.36653061224489797, 0.6261224489795918

t, U, avg_xij: 3, -0.8081632653061225, 0.92

t, U, avg_xij: 4, -1.0, 1.0

Wnioski

Z symulacji wynika, że często w całym procesie coraz więcej relacji między aktorami zaczyna być ze sobą zgodnych, czasami z dosyć mocnym obrotem w stronę skłócenia (aby często wrócić na bardziej pozytywne tory). Ostatecznie (najczęściej) aktorzy szybko dzielą się na zaprzyjaźnione grupy, ale skłócone ze sobą. Im więcej jest aktorów zgodnych, tym większe prawdopodobieństwo o całkowitą zgodę w populacji. Wynika też z tego fakt, że osoby/ grupy, które wyzbyły się dysonansu poznawczego i ostatecznie mające np. podobne opinie będą się wzajemnie przyciągać, a o przeciwnych będą uważane za wrogie oraz odrzucane, o ile nie przeważają osoby ze sobą zgodne.