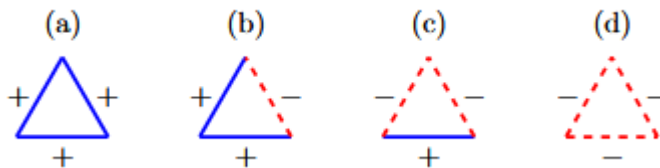


# Model równowagi strukturalnej

Bartosz Gębski

## 1. Opisanie problemu

Teoria równowagi, stworzona przez Fritza Heidera, dotyczy dynamiki pozytywnych i negatywnych relacji w sieci społecznej, podkreślając stabilność i równowagę tych powiązań. Ta teoria ma swoje zastosowanie w różnych dziedzinach, od międzyludzkich relacji aż po stosunki międzynarodowe, gdzie pomaga przewidzieć pewne zachowania w oparciu o równowagę relacji w sieci [1]. Teoria równowagi strukturalnej to pewne rozwinięcie zaproponowane przez Dorwina Cartwright'a i Franka Harary'ego [2]. Teoria równowagi dzieli relacje na przyjazne i wrogie. Podkreśla również wagę triad w zachowywaniu równowagi, gdzie relacje między trzema aktorami mogą prowadzić do stabilności lub napięcia [3]. Istnieją cztery możliwe triady, z czego dwie są zbalansowane według teorii Heidera:



Rysunek 1: Triady oddziaływań zbalansowanych wg Heidera (a oraz c) i niezbalansowanych (b oraz d). Kolorem czerwonym zaznaczona relacja wroga, niebieskim relacja przyjacielska [4]

Zbalansowany układ może przybrać dwie postaci: w jednej aktorzy dzielą się na pewne grupy, które wzajemnie się nie lubią. W drugiej osiąga się tzw. stan raju, co oznacza, że wszystkie relacje w sieci są przyjacielskie. Balansowanie układu odbywa się poprzez zmiany relacji w poszczególnych triadach aktorów, doprowadzając poszczególne triady do stanu równowagi.

## 2. Definicja modelu

Rozważano sieć 50 aktorów, każdy połączony z każdym. Wszystkim parom przypisano losowo relację przyjacielską (zapisaną jako 1) lub wroga (zapisaną jako -1), co zapisywano w macierzy 50x50, gdzie element (x,y) odpowiada relacji aktora x z aktorem y. Dla elementów macierzy (x,y), gdzie x=y (czyli relacja wobec samego siebie) relacje zapisywano w macierzy jako 0.

Przyjmując aktorów  $i, j, k$  gdzie  $x_{ij}$  oznacza relację między aktorami  $i$  oraz  $j$ , to, czy triada jest zbalansowana czy nie, obliczane jest poprzez wymnożenie relacji tych trzech aktorów. Zbalansowane triady dają wynik 1, niezbalansowane -1. Gdy chcemy określić w chwili czasowej  $t$ , jaka poszczególna relacja powinna być w następnym kroku czasowym, posługujemy się wzorem:

$$x_{ij}(t+1) = \text{sign}(x_{ij}(t)x_{jm}(t) + x_{ik}(t)x_{kn}(t))$$

W przypadku, gdy krawędź  $ij$  należy do więcej niż jednej triady, sumuje się wszystkie iloczyny par wartości relacji w każdej triadzie, do której krawędź  $ij$  należy. Gdy wynik jest ujemny, poszukiwana relacja powinna być wroga. Jeśli dodatni, przyjazna. W rzadkich przypadkach wynik może równać się 0 - założono, że wtedy pozostawia się relację bez zmian.

W przypadku istnienia więcej niż jednej triady w sieci, zbalansowanie całego układu osiąga się kiedy  $U = -1$ , co wylicza się ze wzoru:

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

Gdzie  $\Delta$  oznacza liczbę wszystkich możliwych triad w sieci, a  $x_{ij}$  to relacja między aktorami  $i$  oraz  $j$  wynosząca 1 lub -1. Jeśli  $U$  wynosi -1, oznacza to że cały układ jest zbalansowany. Jeśli  $U = -1$  i jednocześnie żadna z relacji nie jest wroga (nie równa się -1), to jest to przypadek tzw. raj.

Wartość  $\langle x_{ij} \rangle$ , oznaczającą średnią wartość relacji w sieci, liczone ze wzoru:

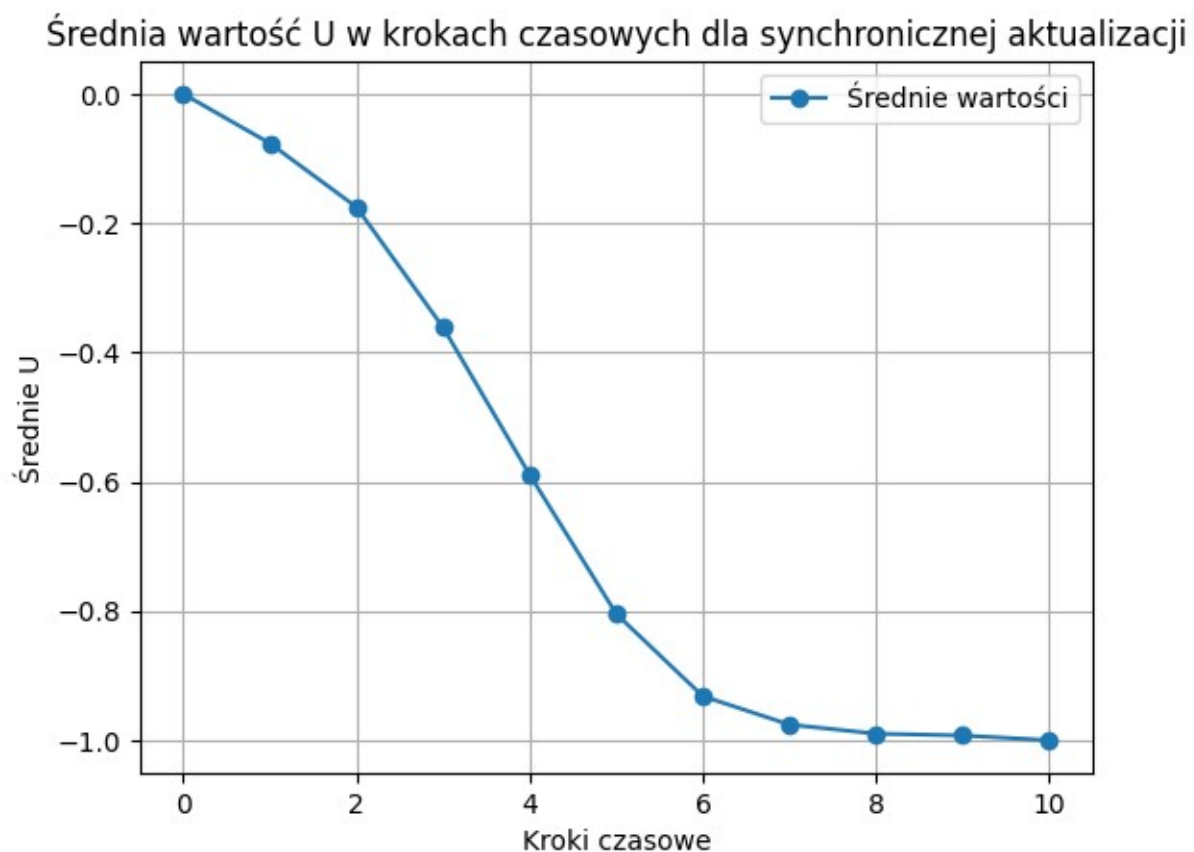
$$x_{ij} = \frac{-1 * a + (b - a)}{b}$$

gdzie  $a$  to liczba wrogich relacji,  $b$  to liczba wszystkich relacji. Wskaźnik ten pomaga wykrywać stan raj przy kończeniu symulacji, kiedy  $x_{ij} = 1$ . Istnieją dwa sposoby aktualizowania relacji: asynchroniczny (aktualizacja natychmiast po wyliczeniu nowej relacji) oraz synchroniczny (po obliczeniu wartości wszystkich relacji są one aktualizowane jednocześnie). Zaimplementowano obie metody, w przypadku asynchronicznego sposobu trzymano się jednej kolejności, od lewej do prawej z dołu w górę.

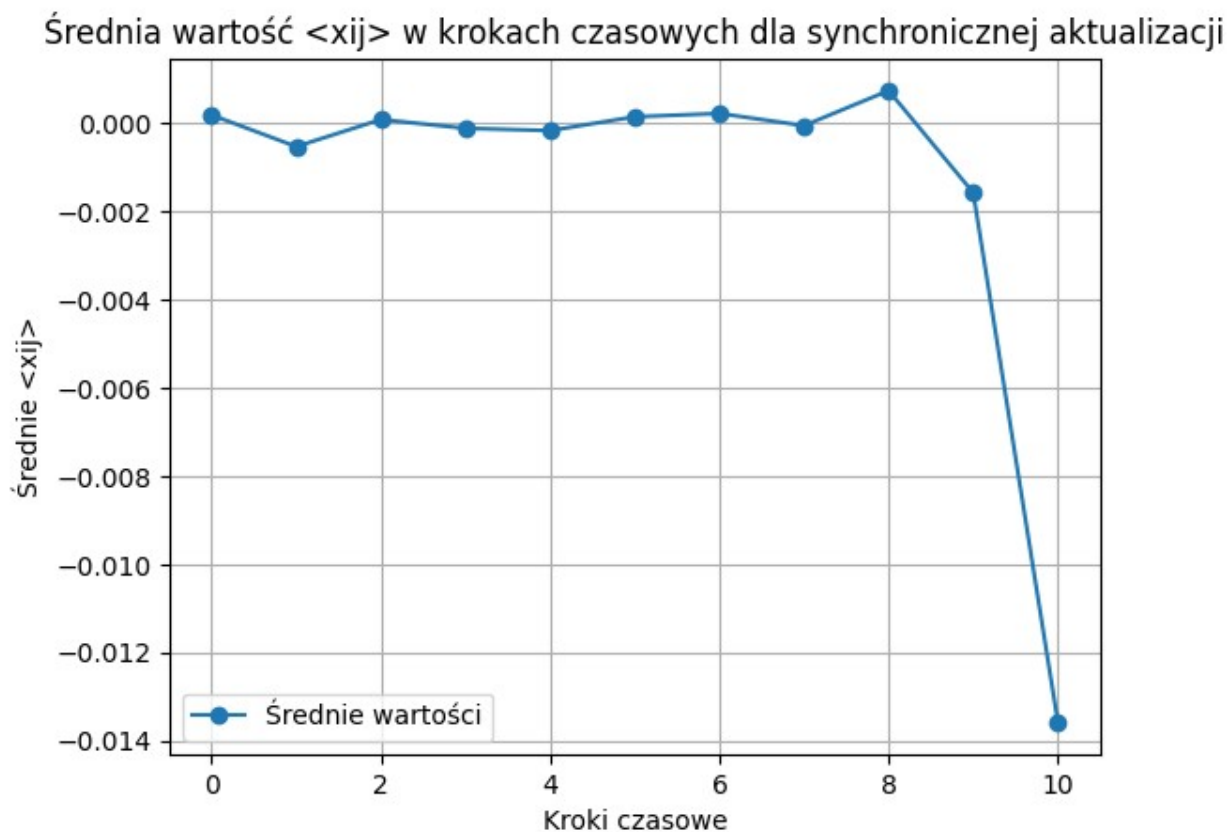
Biorąc pod uwagę możliwość tzw. migotania układu (pętli, w której poszczególne wartości układu zmieniają się w kroku czasowym  $t$ , a w  $t+1$  wracają do poprzednich wartości), zastosowano mechanizm wykrywania takiego przypadku. Wówczas symulacja była restartowana. Zapisane wyniki były eksportowane do programu Gephi, w którym

dokonano wizualizacji.

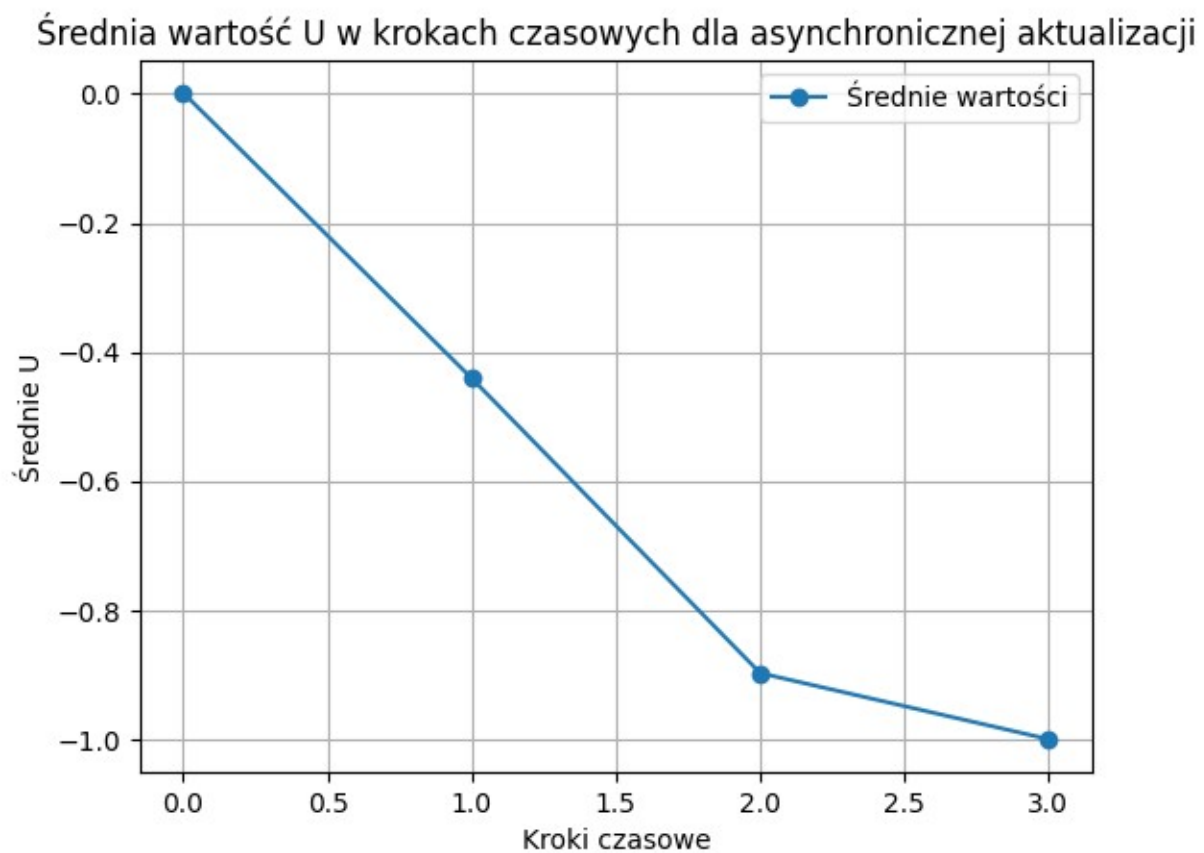
### 3. Wyniki



Rysunek 2: Wykres dla synchronicznej aktualizacji danych przedstawiający uśrednioną wartość U w różnych krokach czasowych. Krok 0 to początkowe ustawienie

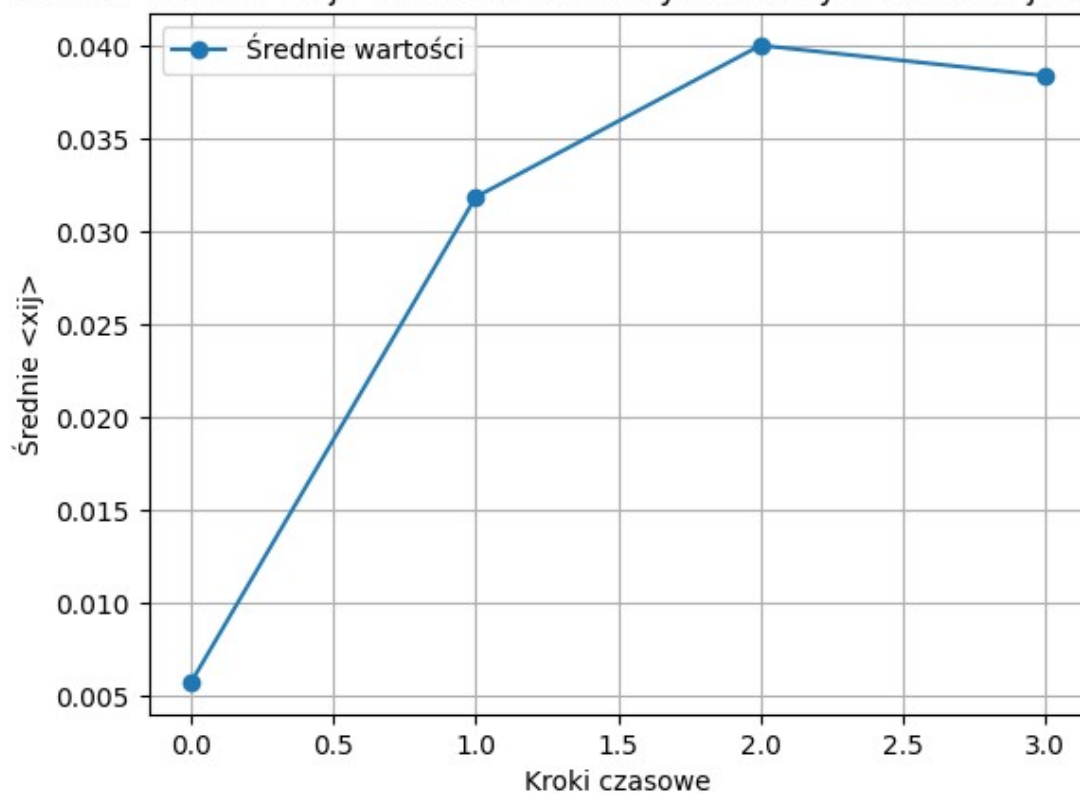


Rysunek 3: Wykres dla synchronicznej aktualizacji danych przedstawiający uśrednioną wartość  $\langle x_{ij} \rangle$  w różnych krokach czasowych. Krok 0 to początkowe ustawienie

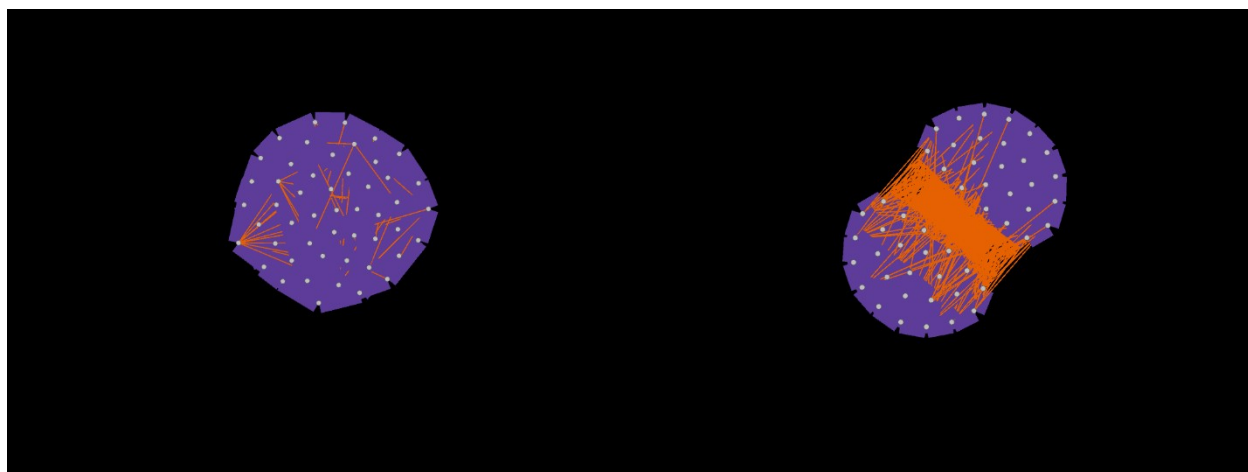


Rysunek 4: Wykres dla asynchronicznej aktualizacji danych przedstawiający uśrednioną wartość  $U$  w różnych krokach czasowych. Krok 0 to początkowe ustawienie

Średnia wartość  $\langle x_{ij} \rangle$  w krokach czasowych dla asynchronicznej aktualizacji

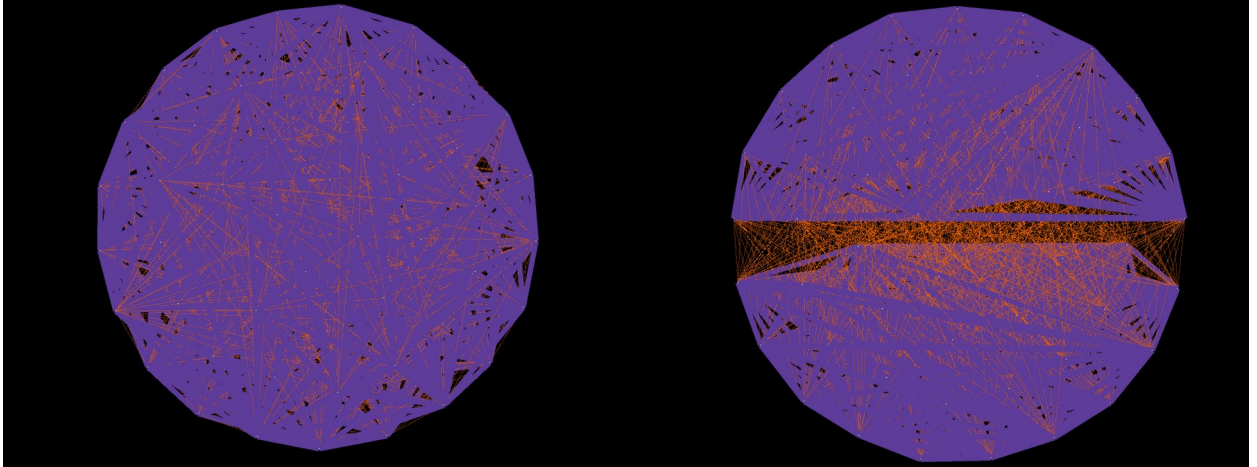


Rysunek 5: Wykres dla asynchronicznej aktualizacji danych przedstawiający uśrednioną wartość  $\langle x_{ij} \rangle$  w różnych krokach czasowych. Krok 0 to początkowe ustawienie

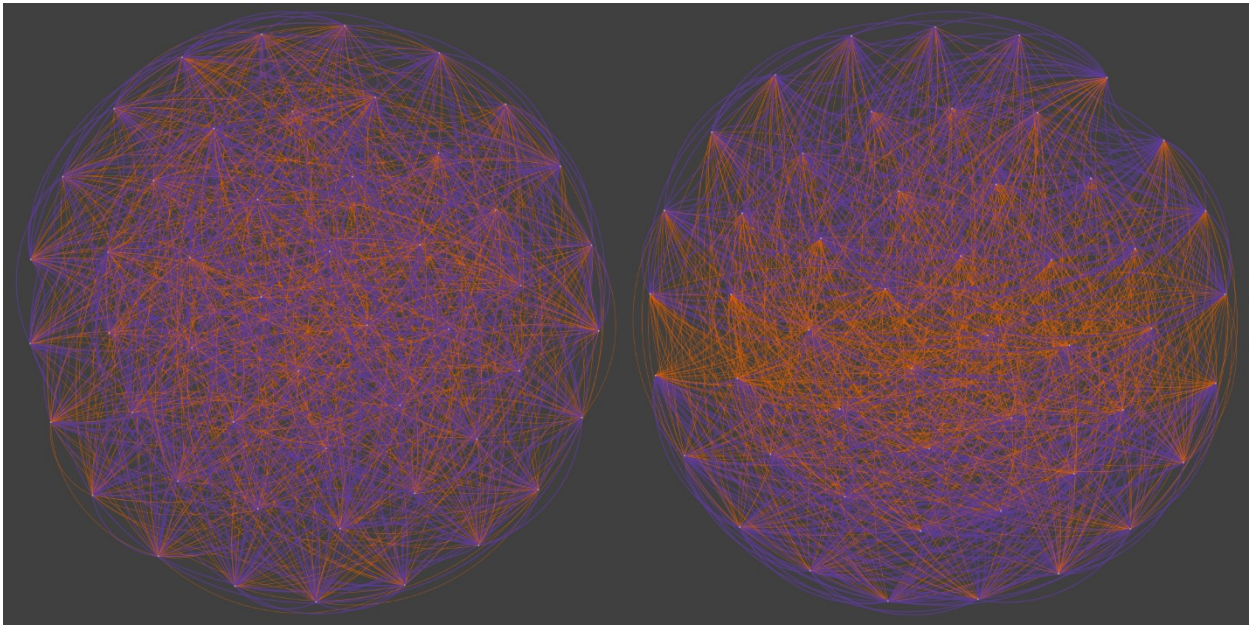


Rysunek 6: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania synchronicznego metodą ForceAtlas. Pomarańczowy – relacja wroga, fioletowy – relacja przyjacielska

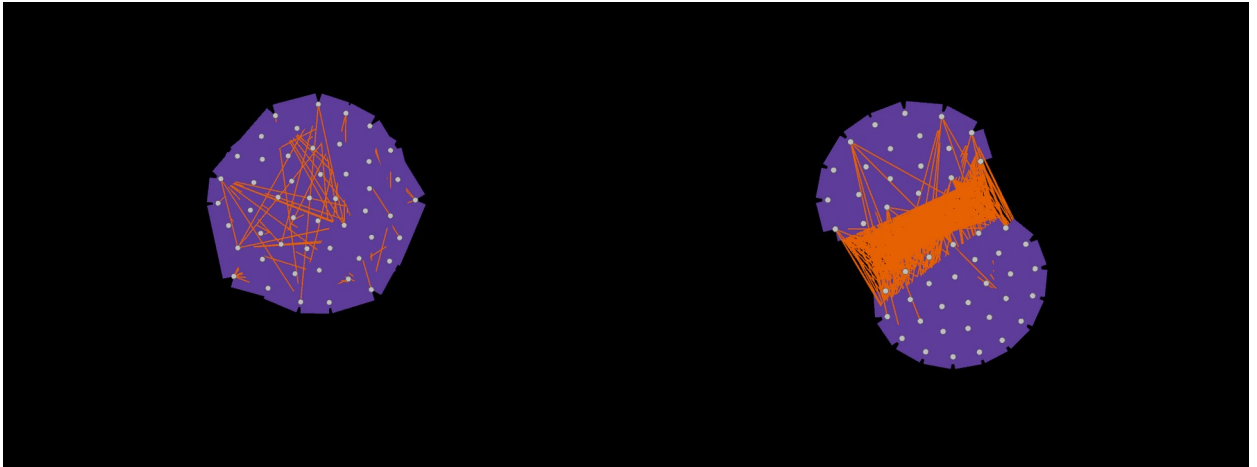




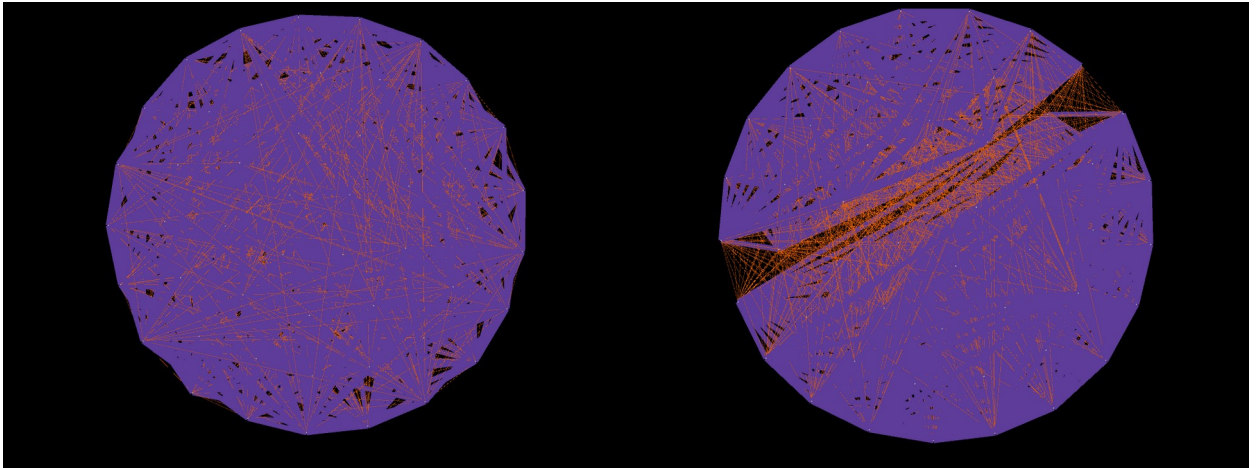
*Rysunek 7: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania synchronicznego metodą Fruchterman Reingold. Pomarańczowy - relacja wroga, fioletowy - relacja przyjacielska*



*Rysunek 8: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania synchronicznego. Pomarańczowy - relacja wroga, fioletowy - relacja przyjacielska*

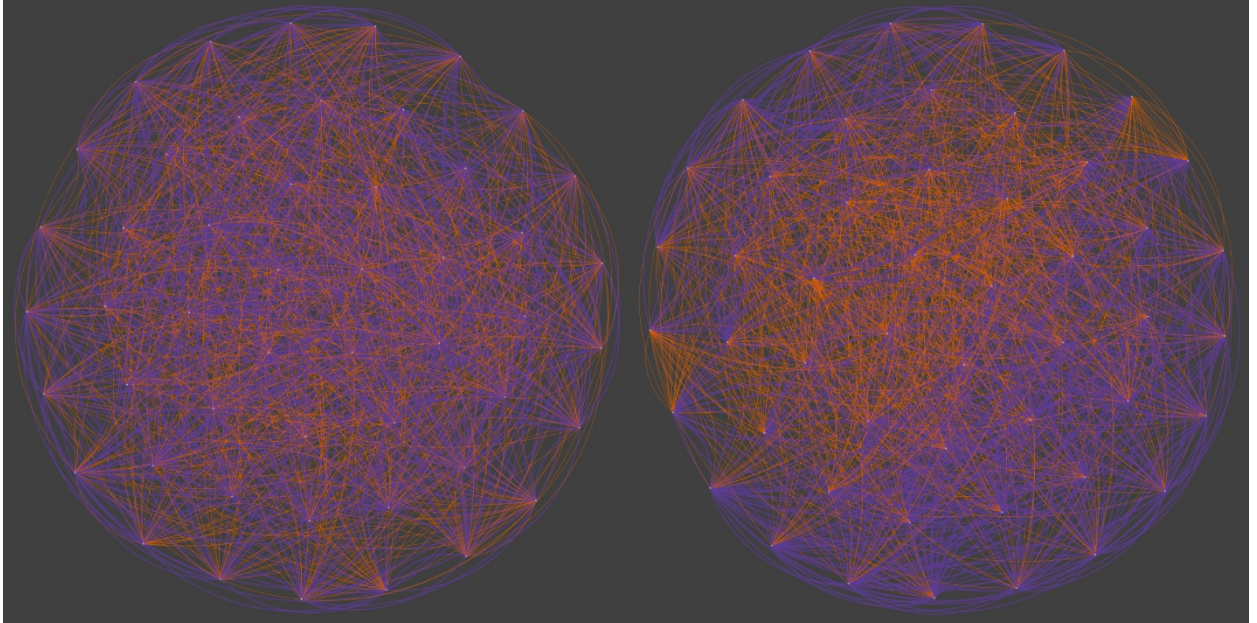


Rysunek 9: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania asynchronicznego metodą ForceAtlas. Pomarańczowy - relacja wroga, fioletowy - relacja przyjacielska



Rysunek 10: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania asynchronicznego metodą Fruchterman Reingold. Pomarańczowy - relacja wroga, fioletowy - relacja przyjacielska





*Rysunek 11: Wizualizacja relacji w początkowym i końcowym kroku czasowym dla aktualizowania asynchronicznego. Pomarańczowy – relacja wroga, fioletowy – relacja przyjacielska*

Wizualizacje i animacje GIF tych przypadków znajdują się w folderze z kodem.

#### 4. **Obserwacje i wnioski**

Na wykresach doskonale widać, że każdy krok czasowy przybliżał nas do osiągnięcia  $U=-1$ . Jest to oczywista rzecz, jednak warto zauważyć, że w przypadku aktualizacji synchronicznej proces ten trwał o wiele dłużej, zajmując nawet 10 kroków czasowych. Gdy mieliśmy do czynienia z asynchronicznym podejściem, ilość kroków czasowych potrzebnych do zbalansowania układu wynosiła maksymalnie 3 kroki. Widać więc, że sposób asynchroniczny jest szybszy w balansowaniu sieci. Jeśli popatrzeć na średnią wartość  $\langle x_{ij} \rangle$ , dla obu sposobów nie były one zbyt wysokie lub niskie, mieszcząc się w przedziale  $\langle -0.015, 0.04 \rangle$ . Jednocześnie wartości dla synchronicznej aktualizacji oscylowały bardzo blisko zera w porównaniu z metodą asynchroniczną. Oddalały się od niego dopiero w końcowych krokach, których zapewne zdarzało się mniej, więc ich średnia wartość nie była tak wiarygodna jak wcześniejszych kroków. Świadczy to więc o tym, że w układach zbalansowanych często przeważają lekko relacje jednej kategorii, jednak jest to przewaga dość nieznaczna. Wizualizacje obu sposobów wyraźnie wskazują na stopniowe wyłanianie

się grup wewnętrznie zaprzyjaźnionych, jednak wzajemnie wrogich. Grupy te nie muszą zawierać tej samej ilości aktorów. Wykonano ponad milion powtórzeń zarówno dla synchronicznego jak i asynchronicznego sposobu aktualizowania relacji. W czasie eksperymentów ani razu nie osiągnięto stanu raju, co nasuwa wniosek, że jest on niezwykle rzadkim przypadkiem. Jednocześnie jest on całkowicie możliwy, ponieważ przy testach z mniejszą liczbą aktorów w sieci stan ten udawało się zaobserwować. W przypadku raju obserwowana wizualizacja sieci składałaby się całkowicie z relacji przyjacielskich, przypuszczalnie więc stopniowo relacje wrogie zanikałyby na rzecz przyjaznych, tworząc jedną, wspólną grupę wewnętrznie zaprzyjaźnioną. Co jakiś czas mechanizm wykrywający migotanie uruchamiał się, zatrzymując symulację – wykrył, że w ostatnich paru krokach stan relacji powtórzył się. Z tego powodu można wywnioskować, że sytuacja taka, chociaż nie zdarza się bardzo często, co jakiś czas może mieć miejsce.

## 5. Źródła

- [1] Hexmoor, H. (2015). *Chapter 4 – Balance Theory* (pp. 31–32). <https://doi.org/10.1016/B978-0-12-800891-1.00004-4>
- [2] Balance theory, Wikipedia.org [dostęp 17.06.2025] [https://en.wikipedia.org/wiki/Balance\\_theory](https://en.wikipedia.org/wiki/Balance_theory)
- [3] Aref, S., Dinh, L., Rezapour, R., & Diesner, J. (2020). Multilevel structural evaluation of signed directed social networks based on balance theory. *Scientific Reports*, 10(1), 15228. <https://doi.org/10.1038/S41598-020-71838-6>
- [4] Krzysztof Malarz, Maciej Wołoszyn; Expulsion from structurally balanced paradise. *Chaos* 1 December 2020; 30 (12): 121103. <https://doi.org/10.1063/5.0022922>