

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Fizyki i Informatyki Stosowanej

KATEDRA INFORMATYKI STOSOWANEJ I FIZYKI KOMPUTEROWEJ



**PRACA INŻYNIERSKA**

**ERNEST JĘCZMIONEK**

**SYMULACJE EWOLUCJI KOALICJI MIESZANYCH**

PROMOTOR:

prof. dr hab. Krzysztof Kułakowski

Kraków 2017

## **OŚWIADCZENIE AUTORA PRACY**

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

.....

PODPIS

**AGH**  
**University of Science and Technology in Krakow**

---

Faculty of Physics and Applied Computer Science  
DEPARTMENT OF APPLIED INFORMATICS AND COMPUTATIONAL PHYSICS



**BACHELOR OF SCIENCE THESIS**

**ERNEST JĘCZMIONEK**

**SIMULATIONS OF EVOLUTION OF MIXED COALITIONS**

SUPERVISOR:  
Professor Krzysztof Kułakowski

Krakow 2017

Serdecznie dziękuję ... tu ciąg dalszych  
podziękowań np. dla promotora, żony,  
sąsiada itp.

## Spis treści

<b>1. Wprowadzenie</b>	6
1.1. Cele pracy	6
1.2. Zawartość pracy	6
<b>2. Opis teoretyczny</b>	7
2.1. Gra 3 osobowa	7
2.2. Model partii	7
2.3. Równania standardowe	7
2.4. Równania replikatorów	8
2.5. Normalizacja prawdopodobieństwa	8
<b>3. Implementacja symulacji</b>	9
3.1. Środowisko QT	9
3.2. OpenGL	9
3.3. Makefile	9
3.4. Jak uruchomić?	9
3.5. Schemat programu	9
<b>4. Wyniki</b>	10
4.1. N gier 3-osobowych niezależnych	10
4.2. N gier 3-osobowych zależnych	10
<b>5. Podsumowanie</b>	12

# 1. Wprowadzenie

Teoria gier wielu osobom kojarzy się z opisem gier towarzyskich między dwojgiem graczy, lecz takie rozgrywki to rzadkość w naszym zróżnicowanym świecie, gdzie zwykle w grę ekonomiczną, społeczną czy polityczną angażuje się wiele uczestników. W niniejszej pracy weźmiemy na tapet jeden z jej filarów, czyli gry  $n$ -osobowe. W tym typie gier ważnym elementem strategii jest odpowiedni wybór koalicjantów. Oczywiście nie będziemy w stanie uwzględnić wszystkich czynników mogących mieć wkład do gry, ale przeanalizujemy dwa równania ewolucyjne, które mogłyby sterować graczami. Zaczynamy ...

## 1.1. Cele pracy

Celem poniższej pracy jest przeprowadzenie symulacji koalicji mieszanych oraz weryfikacji przewidywanych wyników.

## 1.2. Zawartość pracy

Rozdział *Opis teoretyczny* wyjaśnia co to jest gra 3-osobowa i o jakiej tabeli wypłat mówimy. Opisuje typy równań użyte do wyboru koalicjanta. Opis narzędzi, technik, schematy oraz sposób uruchomienia programów znajdziemy w rozdziale *Implementacja symulacji*. Na końcu omówimy rezultaty pracy wraz z rozpatrzeniem szczególnych przypadków.

!!!TO BE DONE!!! [Now06] [HS98] [P.01] [Qt] [Tut] [Sza]

## 2. Opis teoretyczny

### 2.1. Gra 3 osobowa

**O jakiej grze mówimy** W niniejszej pracy będziemy skupiali się na grze 3-osobowej. Będziemy rozpatrywać tylko przypadki w których koalicja dwóch graczy wygrywa. Nie bierzemy pod uwagę sytuacji w których współpraca ze sobą wszystkich graczy mogłaby przynieść najlepsze korzyści. W tabelce wypłat rozważanej gry nie może pojawić się punkt równowagi wynikający ze strategii czystych. W rozważanej przez nas grze celem gracza nie jest zdobycie jak największego zysku, lecz osiągnięcie jak największej liczby wygranych. Osoba znajdująca się poza koalicją przegrywa i wszyscy gracze mają taką samą wagę wyboru.

Będziemy rozważali dwa przypadki: zależnej i niezależnej rozgrywki. W pierwszej części skupimy się na partiach niezależnych, a następnie wykonamy symulacje gier ze sobą powiązanych. W grach zależnych każdy gracz będzie rozgrywał partie z dwoma innymi graczami siedzącymi po obu jego stronach, będzie to przypominało grę w okręgu gdzie każdy z graczy może wybrać czy wstępuje w koalicję z swoim partnerem po prawej czy lewej stronie. W takiej sytuacji nie będzie możliwe granie z oboma partnerami. Będziemy w końcu zmuszeni wybrać z kim trzymamy stałą koalicję, a kogo odrzucimy. Wykluczamy możliwość jakiegokolwiek komunikacji między graczami poza obserwowaniem ich poprzednich zagrań. Model ten z pewnością będzie dużo bardziej dynamiczny, gdyż częstotliwości wybierania sojuszy będzie musiała zmienić zachowanie sojuszników naszych partnerów.

### 2.2. Model partii

OPISZ TROJKAT GRY Z PRAWD I ZROB RYSUNEK

### 2.3. Równania standardowe

$$p_i+ = \alpha \cdot \left(1 - \frac{n_R}{nr_{games}} - \frac{n_L}{nr_{games}}\right) \quad (2.1)$$

## 2.4. Równania replikatorów

$$p_i+ = \alpha \cdot (p_i \cdot (1 - p_i)) \cdot \left(1 - \frac{n_R}{nr_{games}} - \frac{n_L}{nr_{games}}\right) \quad (2.2)$$

## 2.5. Normalizacja prawdopodobieństwa

Jak zauważyliśmy powyższe równania w łatwy sposób mogą wyjść poza przedział  $< -1, 1 >$ . Aby temu zapobiec każda inkrementacja prawdopodobieństwa musi być obłożona funkcją normalizującą. Zdecydowałem się użyć następującej funkcji:

$$norm(p_i) = \begin{cases} 1 & \text{jeżeli } p_i > 1 \\ p_i & \text{jeżeli } 1 \geq p_i \geq -1 \\ -1 & \text{jeżeli } p_i < -1 \end{cases}$$

Zapewne naszą uwagę przykuł także parametr  $\alpha$  obecny w powyższych równaniach. W pierwszym z nich zdecydowałem się na



## **3. Implementacja symulacji**

W tym rozdziale chciałbym przedstawić technologie i narzędzia użyte do wykonania symulacji oraz sposoby ich uruchomienia.

### **3.1. Środowisko QT**

Zdecydowałem się wykorzystać QT Creator IDE z kilku powodów, które zamierzam zaraz rozwinąć. Najważniejszą cechą środowiska jest udostępnienie go na kilku rodzajach licencji. Osobiście użyłem licencji LGPL, która pozwoliła mi bez ponoszenia kosztów korzystać ze środowiska. Kolejnym ważnym elementem jest multiplatformowość pozwalająca w łatwy sposób przenosić kod program między systemami operacyjnymi, o ile nie zostały użyte biblioteki dostępne tylko na jeden z systemów. Kolejną zaletą jest łatwy i intuicyjny interfejs tworzenia graficznego interfejsu użytkownika, osoba mająca wcześniej styczność z chociażby biblioteką Swing Java'y nie powinna mieć problemu z zaadaptowaniem się do formularza QT Creatora. Wykorzystywany jest model sygnałów i slotów, polegający na emitowaniu sygnału przez zdarzenie, który następnie trafia do podłączonego slotu. Jest to w stanie znacznie ułatwić komunikację między elementami. Używanie nowoczesnego języka C++ (ja używałem wersji 14) nie sprawia problemów, lecz powinniśmy być świadomi że przykładowe uruchomienie wątków w aplikacji powinno być zrobione przy użyciu klas i funkcji z biblioteki QT.

### **3.2. OpenGL**

### **3.3. Makefile**

### **3.4. Jak uruchomić?**

### **3.5. Schemat programu**

## 4. Wyniki

### 4.1. N gier 3-osobowych niezależnych

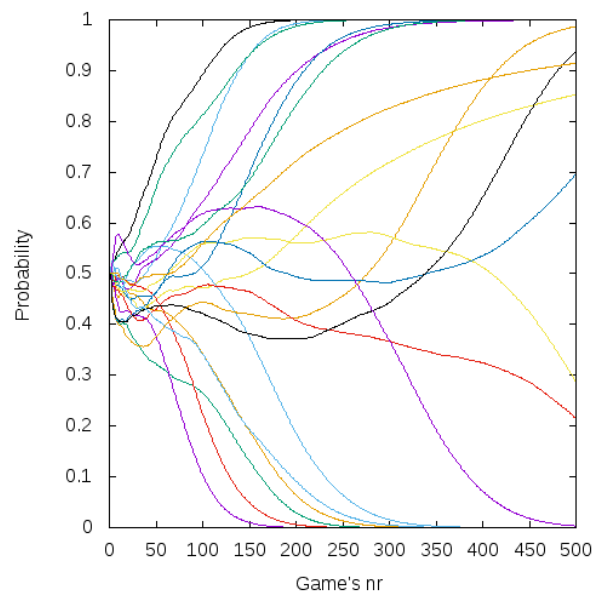
Równania standardowe

Równania replikatorów

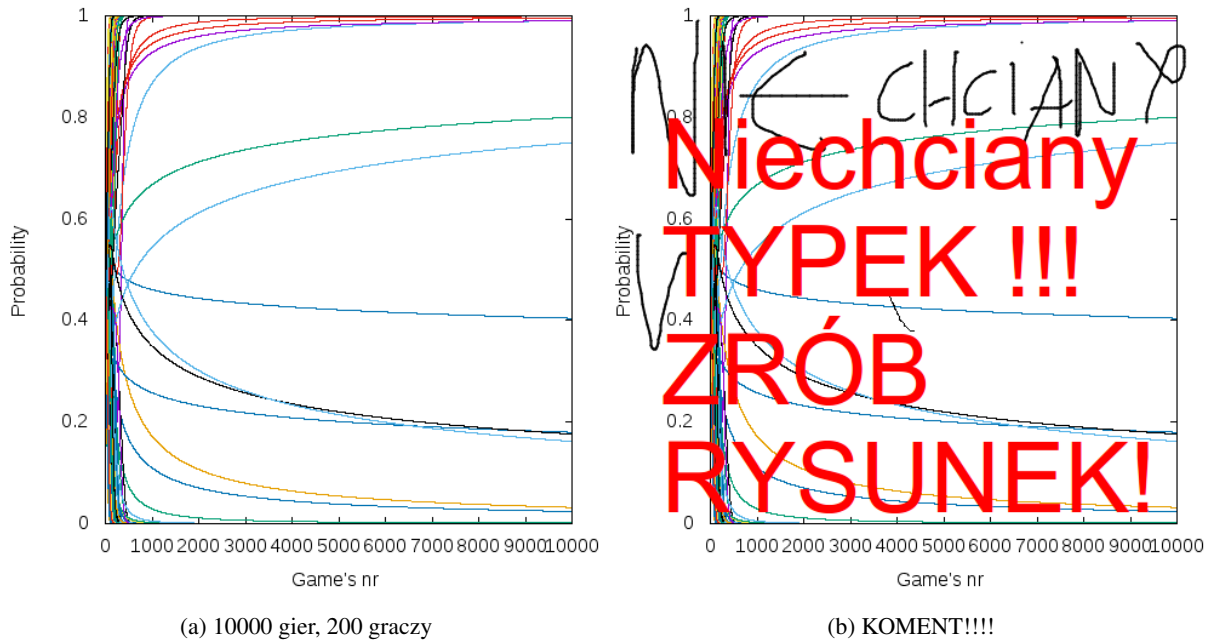
### 4.2. N gier 3-osobowych zależnych

Chciałbym teraz przeanalizować wyniki jednej z symulacji 4.1. Jak już wcześniej zaobserwowaliśmy równania replikatorów dają dużo mniejszą dynamikę decyzji graczy.

Peleton graczy tworzy stabilne koalicje około 300 partii które nie są w stanie ulec zmianie. Pozostałe przypadki tworzą niestabilne koalicje, które zmieniają się w czasie. Najlepszym tego przykładem jest gracz który początkowo gra w kierunku swojego prawego sąsiada, a później zapewne przez jego niechęć po kilku fluktuacjach zaczyna drastycznie zmieniać partnera swojej gry - zaznaczonych kolorem fioletowym. Czynniki losowe graczy utrudnia grę tylko z jednym wybranym partnerem, dlatego jak widzimy podczas pierwszych 50 gier dochodzi do dużej liczby zmian zachowań graczy. Szczególnie widoczne jest to w pierwszych 50 partiach, gdzie gracze dopiero szacują zachowanie sąsiadów. W kolejnych 50 partiach gracze zachowują się coraz bardziej liniowo, gdyż błąd przewidywanego i realnego prawdopodobieństwa przeciwnika spada. Nie wchodzący w stałą koalicja mogą należeć do łańcucha graczy niezdecydowanych lub jednostek znajdujących się pomiędzy dwoma silnymi koalicjami które nie dają szansy na przyłączenie się do żadnej. Warty uwagi jest fakt że ostatnia monotoniczności funkcji zachodzi dopiero około 300 gry.



Rysunek 4.1: 500 gier, 20 graczy



Rysunek 4.2: Długa gra

Rozpatrzmy teraz dużo dłuższą grę w której zaangażowanych jest więcej graczy, co może pokazać nam przypadki szczególne. Na rysunku widzimy że większość zawodników osiąga stabilne koalicje przed grą 1000. Widzimy grupę kilku graczy którzy pomimo tak dużej ilości gier nie byli w stanie zawiązać trwałych koalicji.

CHECK Mogło to wynikać z dwóch faktów. Po pierwsze mogli znaleźć się między graczami znajdującymi się w trwałe w sojuszach którzy nie byli zainteresowani wchodzeniem w nowe. Drugą przyczyną może być nieznaną prawdziwego prawdopodobieństwa podejmowania decyzji przez sąsiadów które jest tylko wartością znaną z rozgrywki różnica pomiędzy faktycznym prawdopodobieństwem gry sąsiada a tym co reprezentował w rozgrywce może się znacząco różnić wpływając na Mylną ocenę prawdopodobieństwa gry zawodników i mogących ustalić 1a stabilnej koalicji rysunek 3 pokazuje przypadek w którym dwóch silnych policjantów nie jest zainteresowanych wejście w sojusz z osamotnionych zawodnikiem

## **5. Podsumowanie**

To be done!

## Spis rysunków

4.1	500 gier, 20 graczy . . . . .	10
4.2	Długa gra . . . . .	11

## Bibliografia

- [HS98] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge, 1998.
- [Now06] M. A. Nowak. *Evolutionary dynamics: exploring the equations of life*. The Belknap press of Harvard university press, Cambridge, Massachusetts and London, England, 2006.
- [P.01] Straffin P. *Teoria gier*. Wydawnictwo Naukowe SCHOLAR, Warszawa, 2001.
- [Qt] Qt main site. <http://doc.qt.io/>. 2017-09-01.
- [Sza] Msc/eng thesis template of university of science and technology in krakow (agh). <https://www.sharelatex.com/templates/thesis/agh>. 2017-09-01.
- [Tut] Qt5 tutorial opengl with qglwidget - 2017. [http://www.bogotobogo.com/Qt/Qt5\\_OpenGL\\_QGLWidget.php](http://www.bogotobogo.com/Qt/Qt5_OpenGL_QGLWidget.php). 2017-09-01.