

# Algorytm Minimax dla gry w Warcaby

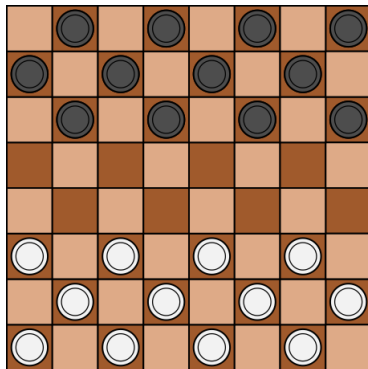
**Janusz Witkowski**

Praca napisana pod kierunkiem **dra Macieja Gębali**

Grudzień 2022, Wrocław

# Warcaby

Warcaby są grą dwuosobową o doskonałej informacji i sumie zerowej. W warcabach gracze na zmianę poruszają się dwoma różnymi typami figur (pionami i damkami) po ukosach, a celem gry jest uniemożliwienie ruchu przeciwnikowi.



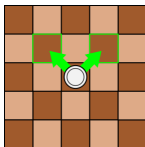
**Rysunek:** Plansza startowa w warcabach

# Wariant angielski

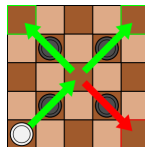
Praca rozpatruje szczególną wersję warcabów - **wariant angielski**.

Wariant ten wprowadza dwie zmiany:

- ▷ Piony nie mogą poruszać się do tyłu,
- ▷ Damki nie ruszają się na dystansy większe niż jedno pole.



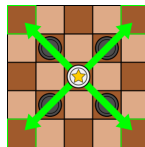
Rysunek: Ruchy dla piona



Rysunek: Bicia dla piona

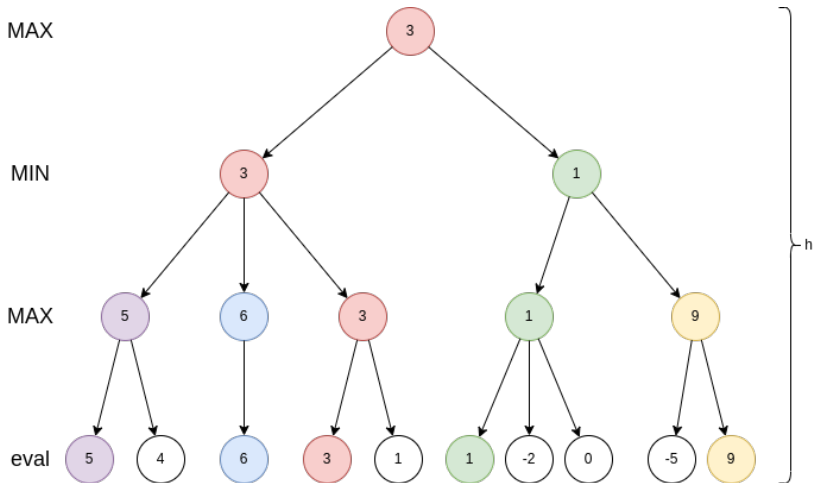


Rysunek: Ruchy dla damki



Rysunek: Bicia dla damki

# Minimax (przykład)



**Rysunek:** Przykładowe wywołanie algorytmu Minimax. W podanym drzewie wierzchołkami są stany rozgrywki, do których przypisuje się wartości ewaluacji.

# Funkcja oceny heurystycznej

$$F(S) = \sum_{i=1}^n param_i(S) * weight_i$$

<i>i</i>	<i>param</i>	<i>weight</i>
1	Liczba sojusznicznych pionów	15
2	Liczba przeciwnych damek przy ścianie	-123
3	Liczba możliwych ruchów gracza	0
4	Czy pion przeciwnika znajduje się w kącie	47
...	...	...

**Tablica:** Przykładowe parametry z przykładowymi wagami.

# Algorytm genetyczny

- ▶ Celem pracy jest wyznaczenie jak najlepszego ciągu wag do funkcji oceny heurystycznej.
- ▶ Pomysł: zastosowanie algorytmu genetycznego.

**Algorytm genetyczny** symuluje dobór naturalny w przyrodzie. Na populacji osobników (zbiorze rozwiązań problemu) wykonuje się operacje:

- 1 Ewaluacja osobników
- 2 Selekcja zbioru rodziców
- 3 Krzyżowanie
- 4 Losowe mutacje
- 5 Tworzenie nowej populacji

# Język i struktura

Projekt został napisany w języku **JAVA** w wersji *OpenJDK 17.0.4* (choć wykorzystuje funkcjonalności z *OpenJDK 14*).

Klasa **MinMax** przeszukuje przestrzeń stanów stworzoną z obiektów klasy **State**.

Klasa **Genetic** przeprowadza selekcje na tablicach liczb całkowitych oraz operuje na plikach populacji i plikach ciągów wag w katalogu **heuristics**.

# Programy wywoławcze

Poniższe programy wywołuje się z linii poleceń z odpowiednimi argumentami.

## ▶ Play

- Rozpoczyna rozgrywkę między graczami (każdy z nich może być człowiekiem lub komputerem)
- Dla graczy komputerowych należy podać również pliki z ciągiem wag
- I/O na poziomie konsoli

## ▶ Find

- Rozpoczyna sesję algorytmu genetycznego
- Może wznowić przedwcześnie przerwana sesję z pliku populacji
- Wynik zapisuje w **heuristics/output/**

## ▶ Show

- Nazwa pliku osobnika do wyświetlenia



# Wyniki: wagi parametrów

**Cel eksperymentu:** Poznać względne wagi wszystkich parametrów; sprawdzić które parametry nie mają znaczenia w rozgrywce (waga bliska zero).

**Wyniki:**

- ▶ Najważniejszymi parametrami w rozgrywce okazały się  $X$ ,  $Y$ ,  $Z$  (*omówienie*).
- ▶ Najgorzej punktowane parametry to  $A$ ,  $B$ ,  $C$  (*omówienie*).
- ▶ Najmniejszy wpływ na grę mają parametry  $U$ ,  $V$ ,  $W$  (*omówienie*). Będzie można opuścić te parametry w następnych eksperymentach.

# Wyniki: różne głębokości

**Cel eksperymentu:** Sprawdzić czy puszczenie dwóch sesji algorytmu genetycznego z różnymi głębokościami przeszukiwań (odpowiednio 4 i 5) dają różne rezultaty.

**Wyniki:** *TBA.*

# Dalszy rozwój projektu

Propozycje rozszerzenia projektu:

- Walka z efektem horyzontu
- Badania perspektyw MINa i MAXa
- Silnik do analizy rozgrywek

- ▷ L. Pijanowski. *Przewodnik gier*. Iskry, 1978.
- ▷ J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, S. Sutphen. *Checkers is solved*. 2007.
- ▷ J. Mańdziuk, M. Kusiak, K. Walędzik. *Evolutionary-based heuristic generators for checkers and give-away checkers*. 2007.
- ▷ S. Russel, P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc., 2010.

Dziękuję za uwagę.

# Implementacja algorytmu genetycznego

## 1 Osobniki

Ciągi wag w postaci tablic liczb całkowitych.

## 2 Ewaluacja

Pojedynek każdy-z-każdym (białe/czarne i na odwrót)

## 3 Selekcja

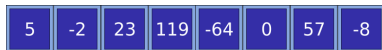
Ruletka (lepiej grające osobniki mają większą szansę na przejście).

## 4 Krzyżowanie

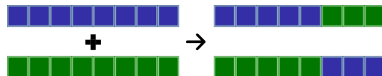
Dzieci powstają poprzez wymieszanie cech rodziców.

## 5 Mutacje

Szansa na wylosowanie nowej wartości losowej wagi w ciągu.



Rysunek: Model poglądowy osobnika



Rysunek: Model krzyżowania