

Warcaby klasyczne

1. Wstęp teoretyczny

Teoria gier (teoria racjonalnych zachowań) to gałąź matematyki dotycząca zachowania w sytuacjach konfliktowych (strategicznych), gdzie wynik uczestnika zależy od wyborów podejmowanych przez innych uczestników i przez niego samego. Wśród gier możemy wyróżnić kilka typów. Pierwszym typem gier, dla których podjęto próby ich rozwiązania były gry dwuosobowe u sumie stałej.

Warcaby klasyczne zaliczamy do gier dwuosobowych (w grze bierze udział dwóch graczy), skończonych (każda rozgrywka musi się kiedyś skończyć), z pełną informacją (obaj gracze mają dokładną wiedzę o sytuacji w grze), o sumie zerowej (cele graczy są dokładnie przeciwnie, wypłaty wszystkich graczy zdeterminowane przez wynik gry sumują się do zera). [1]

W 1963 roku ogromną sensację wzbudził program Arthurusa Samuela, napisany na maszynie IBM, który pokonał w grze w warcaby wybitnego gracza Roberta Neaelya. Po słynnym pojedynku problem warcabów został uznany za rozwiązany i zainteresowanie nim zmalało. Dopiero w latach dziewięćdziesiątych 1989 roku Jonathan Schaeffer opracował nowy algorytm. „Chinook” już po dwóch miesiącach udoskonalania wygrał warcabową olimpiadę dla programów w Londynie.

Analizował on konsekwencje każdego posunięcia na 27 ruchów do przodu. Jego autem była dopracowana do perfekcji końcówka, korzystał on bowiem z bazy danych, która zawierała skutki wszystkich możliwych ruchów w sytuacji, kiedy na planszy pozostały już ostatnie piony – początkowo cztery, ale z czasem więcej. Dołożenie każdego kolejnego pionu potęgowało jednak liczbę możliwych końcowych posunięć. W starciach z najwybitniejszymi graczami algorytm popełniał błędy i przegrywał.

W 2001 roku, kiedy pojawiły się szybsze generacje procesorów, Schaeffer powrócił do swoich badań i osiągnął sukces. Rozszyfrował wszystkie 40 bilionów scenariuszy gry z dziesięcioma pionami do planszy. Opracowany przez niego program nie popełnia błędów. Jeżeli człowiek wybierze wszystkie najlepsze posunięcia, jest w stanie doprowadzić co najwyżej do remisu, ale nigdy nie do wygranej. Wynik nie zależy od tego, czy rozgrywkę zaczynają piony białe czy czarne. [2]

2. Opis wykorzystanych algorytmów sztucznej inteligencji

2.1. Model gry

W celu zastosowania odpowiedniego algorytmu należy odpowiednio sklasyfikować rozpracowywaną grę, czyli określić przyjmowany dalej model gry. Podstawowym założeniem warcabów jest, że w grze bierze udział dwóch uczestników, gracze wykonują ruchy naprzemiennie, w każdej sytuacji na planszy jest skończona liczba możliwych do wykonania ruchów, sytuacja na planszy i wykonany ruch jednoznacznie wyznaczają następną sytuację na planszy, każda możliwa sytuacja na planszy może być jednoznacznie zaklasyfikowana jako wygrana pierwszego gracza, wygrana drugiego gracza, remis lub sytuacja nierozstrzygnięta. Dla tego typu gier można rozważyć intelligentne techniki wyboru ruchu.

2.2. Drzewo gry

Pojedyncza partia gry może być w pełni opisana przez ciąg naprzemiennych ruchów obu graczy, od początkowego ustawienia do rozstrzygnięcia. Aby w dowolnym momencie w trakcie trwania partii wybrać najbardziej odpowiedni ruch dla jednego z graczy, można rozważyć wszystkie możliwe scenariusze jej dalszego ciągu, rozpoczynające się różnymi możliwymi do wybrania ruchami tego gracza, po każdym z których może nastąpić każdy możliwy ruch drugiego gracza, i tak dalej. Naturalną reprezentacją dla takiej przestrzeni sytuacji, możliwych do osiągnięcia po kolejnych ruchach graczy, jest drzewo gry.

Węzły drzewa gry odpowiadają sytuacjom na planszy. W korzeniu drzewa znajduje się węzeł odpowiadający sytuacji, w której poszukujemy najlepszego ruchu dla jednego z graczy nazywanego dalej po prostu graczem. Każdy poziom w drzewie gry jest ściśle związany z informacją, który gracz ma w nim wykonywać ruch. Jeżeli więc poziom korzenia jest poziomem gracza, kolejny będzie poziomem drugiego gracza, nazywanego dalej przeciwnikiem i dalej naprzemiennie. Gałęzie wychodzące z każdego węzła reprezentują wszystkie możliwe ze względu na reguły gry i aktualną sytuację na planszy ruchy odpowiedniego gracza. Każda z tych gałęzi prowadzi do węzła potomnego związanego z kolejną sytuacją na planszy, osiąganą po wykonaniu odpowiedniego ruchu, w której ruch będzie wykonywać drugi z graczy.

Pełne drzewo gry dla danej sytuacji początkowej to takie drzewo, w którym każdy węzeł odpowiadający nierozstrzygniętej sytuacji na planszy ma gałęzie wychodzące odpowiadające wszystkim możliwym ruchom oraz odpowiednie węzły potomne, reprezentujące sytuacje uzyskane po tych ruchach. W pełnym drzewie wszystkie węzły terminalne (liście) reprezentują sytuacje, w których partia gry jest rozstrzygnięta. Ze względu na liczbę możliwych sytuacji w nietrywialnych grach budowanie takiego pełnego drzewa nie jest praktycznie możliwe. Zachodzi w związku z tym konieczność rozważania drzew gry, w których pozostają węzły terminalne odpowiadające nie rozstrzygniętym sytuacjom na planszy.

2.3. Przeszukiwanie drzewa

Wybór ruchu podczas rozgrywki opiera się na przeszukaniu przestrzeni wszystkich możliwych scenariuszy partii w celu znalezienia najbardziej korzystnego. Pierwszy ruch z takiego najkorzystniejszego scenariusza jest uznawany za poszukiwany najkorzystniejszy ruch. Wygodnie jest jednak zastąpić bezpośrednie przeszukiwanie przestrzeni scenariuszy przeszukiwaniem przestrzeni sytuacji na planszy, z zachowaniem kolejności przechodzenia między nimi. Ciąg kolejnych sytuacji wyznacza wówczas scenariusz. Przeszukiwaną przestrzeń sytuacji na planszy reprezentuje drzewo gry. Jego węzły odpowiadają stanom, czyli poszczególnym sytuacjom na planszy wraz z wskazaniem aktualnego gracza, a krawędzie operatorom, czyli dowolnym możliwym ruchom gracza.

2.4. Strategie minimaksowe

Uczestnictwo w rozgrywce przeciwnika, który autonomicznie wybiera swoje ruchy i którego decyzje nie są z góry znane wymusza zastosowanie nieco innych, specjalizowanych algorytmów przeszukiwania. Najczęściej stosowane, najostrożniejsze podejście do ominięcia tej trudności polega na przyjęciu założenia, że przeciwnik dążąc do wygrania partii zawsze wybiera najkorzystniejszy dla siebie, a więc najmniej korzystny dla gracza ruch. Jest to tylko założenie, które dla praktycznych przeciwników, zwłaszcza niezalgoritmizowanych, nie musi być prawdziwe, lecz jest możliwe bezpiecznym rozwiązaniem przy braku znajomości przeciwnika. Jeżeli istniałaby możliwość przypisania węzłom drzewa gry liczbowych ocen odzwierciedlających w pewien sposób szanse wygranej gracza, to na poziomie ruchu gracza należy zakładać wybór maksymalizujący taką ocenę, zaś na poziomie przeciwnika - wybór ją minimalizujący.

2.5. Minimaks

Węzłom terminalnym drzewa, to znaczy odpowiadającym sytuacjom, w których partia jest rozstrzygnięta przypisuje się liczbową ocenę ich użyteczności z punktu widzenia gracza w następujący sposób: ocena dodatnia M w przypadku wygranej gracza, ocena ujemna $-M$ w przypadku wygranej przeciwnika i ocena 0 w przypadku remisu. Ocena dokonywana jest z punktu widzenia gracza, niezależnie od tego, czy oceniany węzeł terminalny drzewa gry znajduje się na poziomie gracza, czy na poziomie przeciwnika. Po dokonaniu w opisany sposób oceny wszystkich węzłów terminalnych drzewa gry, oceny można propagować do węzłów wewnętrznych. Na poziomie gracza węzeł otrzymuje ocenę równą maksimum ocen jego węzłów potomnych, a na poziomie przeciwnika węzeł otrzymuje ocenę równą minimum ocen jego węzłów potomnych.

Efektem stosowania opisanej wyżej metody oceniania węzłów wewnętrznych drzewa gry, nazywanej metodą minimaksową albo zasadą minimaksu, jest przypisanie każdemu węzlowi oceny odzwierciedlającej najlepszy możliwy wynik partii dla gracza, przy założeniu najmniej dla niego korzystnych decyzji przeciwnika. Dla węzłów z poziomu gracza ruch prowadzący do węzła o maksymalnej ocenie jest ruchem optymalnym w sensie zasad minimaks.

2.6. Obcięty Minimaks

Ze względu na ogromną liczbę możliwych sytuacji i scenariuszy stosowanie pełnego algorytmu minimaks nie jest możliwe do wykonania w jakimkolwiek rozsądny czasie i z wykorzystaniem pamięci o jakichkolwiek praktycznie dostępnych pojemnościach. Sposobem dostosowania algorytmu minimaks do wymogów praktycznych jest ograniczenie głębokości analizowania drzewa gry do pewnej liczby poziomów zależnej od złożoności gry i dostępnej mocy obliczeniowej. Oznacza to, że w analizowanym drzewie gry będą węzły odpowiadające nierostrzygniętej partii nie posiadające węzłów potomnych.

Aby zastosowanie algorytmu minimaks było możliwe w niepełnym drzewie gry, każdy z węzłów musi otrzymać ocenę opartą nie na wyniku ani na ocenach węzłów potomnych, lecz wyłącznie na analizie związanego z nim stanu gry. Do oceny węzłów ograniczonego drzewa wykorzystuje się funkcję heurystyczną. Funkcja ta ocenia jakość stanu jedynie na podstawie jego analizy, bez faktycznego generowania jakichkolwiek stanów następnych. Funkcja ocenająca oceniać będzie użyteczność stanu z punktu widzenia gracza bez rozważania dalszych możliwych ruchów, wyłącznie rozważając sytuację na planszy, to znaczy ilość pionków i damek. Zakładając dostępność takiej heurystycznej funkcji oceny, obcięty algorytm minimax można sformułować jako prostą modyfikację rekurencyjnej wersji pełnego algorytmu.

2.7. Heurystyczna ocena węzłów

Skuteczność algorytmów opartych na zasadzie minimaksu w przypadku, gdy przeglądany może być tylko minimalny fragment pełnego drzewa gry w dużej mierze zależy od jakości używanej funkcji heurystycznej. Funkcja ta musi przypisywać ocenę dodatnią stanowi, z którego większe szanse wygranej ma gracz i ocenę ujemną stanowi, z którego większe szanse wygranej ma przeciwnik, przy czym powinna przypisywać ocenę tym większą co do wartości bezwzględnej, im przewaga szans wygranej odpowiednio gracza albo przeciwnika jest większa. W grze w warcaby funkcja taka jest obliczana z uwzględnieniem liczby poszczególnych figur w posiadaniu gracza i przeciwnika.

3. Zasady gry

- 1) Warcaby rozgrywane są na planszy o rozmiarze 8x8. Wiersze i kolumny zostały ponumerowane liczbami z zakresu 0-7. Każde pole na planszy, oznaczone jako [] można więc opisać współrzednymi (a,b), gdzie a to numer wiersza, a b to numer kolumny.
- 2) Gra toczy się jedynie na polach o parzystych numerach kolumn i nieparzystych numerach wierszy oraz nieparzystych numerach kolumn i parzystych numerach wierszy.
- 3) Użytkownik rozpoczyna rozgrywkę z dwunastoma pionami koloru białego, oznaczonymi na planszy jako W i ustawnionymi na dolnej części planszy. Przeciwnik (komputer) dysponuje dwunastoma pionami koloru czarnego, oznaczonymi na planszy jako B i ustawnionymi na górnej części planszy. Piony na początku każdej rozgrywki ustawione są w poniższy sposób:

KOLUMNY									
0	1	2	3	4	5	6	7	0	W
0 [B]	[B]	[B]	[B]	[B]	[B]	[B]	[B]	1 I	
1 []	[]	[]	[]	[]	[]	[]	[]	2 E	
2 []	[B]	[]	[B]	[]	[B]	[]	[B]	3 R	
3 []	[]	[]	[]	[]	[]	[]	[]	4 S	
4 []	[]	[]	[]	[]	[]	[]	[]	5 Z	
5 [W]	[]	[W]	[]	[W]	[]	[W]	[]	6 E	
6 []	[W]	[]	[W]	[]	[W]	[]	[W]	7	
7 [W]	[]	[W]	[]	[W]	[]	[W]	[]		
0	1	2	3	4	5	6	7		

Rys. 1. Początkowe ustawienie pionów na planszy

- 4) Jako pierwszy ruch wykonuje użytkownik. Następnie gracze na zmianę wykonują kolejne ruchy.
- 5) Piony mogą poruszać się o jedno pole do przodu po przekątnej, zajmując wolne pola.
- 6) W celu wykonania ruchu użytkownik podaje współrzędne pola, na którym znajduje się pion, którym ma zostać wykonany ruch. Program pokazuje dostępne możliwości, to znaczy współrzędne pól, na które może zostać przeniesiony wybrany pion. Następnie użytkownik podaje współrzędne pola, na które ma zostać przeniesiony wybrany pion.

	KOLUMNY									
	0	1	2	3	4	5	6	7	0	W
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[W]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
5 2
from: 5 2 to: 4 3
from: 5 2 to: 4 1
Podaj gdzie chcesz sie ruszyc
4 1

	KOLUMNY									
	0	1	2	3	4	5	6	7	0	W
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[W]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Rys. 2.

Prawidłowy sposób wykonania ruchu

7) Pion nie może zostać przeniesiony na pole zajęte przez inny pion. W przypadku próby wykonania takiego ruchu program ponownie poprosi o podanie współrzędnych pola, na które ma zostać przeniesiony wybrany pion. W przypadku wybrania przez użytkownika pionu, który nie ma możliwości ruchu, program wyda stosowny komunikat i ponownie poprosi o podanie współrzędnych pola, na którym znajduje się pion, który ma zostać przeniesiony.

	KOLUMNY									
	0	1	2	3	4	5	6	7	0	W
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[W]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
7 2
Brak mozliwosci ruchu dla pionka, wybierz inny pionek
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:

Rys. 3. Nieprawidłowy sposób wykonania ruchu

	KOLUMNY								
	0	1	2	3	4	5	6	7	0 W
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	[] W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	[] I
2	[]	[]	[]	[B]	[]	[B]	[]	[B]	[] E
3	[]	[]	[B]	[]	[]	[]	[]	[]	[] R
4	[]	[W]	[]	[]	[]	[]	[]	[]	[] S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	[] Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	[] E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	[] 7
	0	1	2	3	4	5	6	7	

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
4 1
from: 4 1 to: 3 0
Podaj gdzie chcesz sie ruszyc
3 2
Podaj gdzie chcesz sie ruszyc

Rys. 4. Nieprawidłowy sposób wykonania ruchu

8) Pion nie może zostać przeniesiony na pole sąsiadujące w pionie lub poziomie (pola o parzystych numerach kolumn i parzystych numerach wierszy lub nieparzystych numerach kolumn i nieparzystych numerach wierszy). W przypadku próby wykonania takiego ruchu program ponownie poprosi o podanie współrzędnych pola, na które ma zostać przeniesiony wybrany pion.

	KOLUMNY								
	0	1	2	3	4	5	6	7	0 W
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	[] W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	[] I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	[] E
3	[]	[]	[]	[]	[]	[]	[]	[]	[] R
4	[]	[]	[]	[]	[]	[]	[]	[]	[] S
5	[W]	[]	[W]	[]	[W]	[]	[W]	[]	[] Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	[] E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	[] 7
	0	1	2	3	4	5	6	7	

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
5 2
from: 5 2 to: 4 3
from: 5 2 to: 4 1
Podaj gdzie chcesz sie ruszyc
4 2
Podaj gdzie chcesz sie ruszyc

Rys. 4. Nieprawidłowy sposób wykonania ruchu

9) Pion nie może zostać przeniesiony do tyłu. W przypadku próby wykonania takiego ruchu program ponownie poprosi o podanie współrzędnych pola, na które ma zostać przeniesiony wybrany pion.

	KOLUMNY									
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[B]	[]	[]	[B]	[]	[]	3	R
4	[]	[W]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
0	1	2	3	4	5	6	7			

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
4 1
from: 4 1 to: 3 0
Podaj gdzie chcesz sie ruszyc
5 2
Podaj gdzie chcesz sie ruszyc

Rys. 5. Nieprawidłowy sposób wykonania ruchu

10) Pion nie może zostać przeniesiony o więcej niż jedno pole.

	KOLUMNY									
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[W]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
0	1	2	3	4	5	6	7			

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
5 2
from: 5 2 to: 4 3
from: 5 2 to: 4 1
Podaj gdzie chcesz sie ruszyc
3 4
Podaj gdzie chcesz sie ruszyc

Rys. 6. Nieprawidłowy sposób wykonania ruchu

11) Po wykonaniu ruchu przez użytkownika, program wyświetla aktualny stan planszy. Następnie automatycznie ruch wykonuje przeciwnik (komputer). Program wyświetla ruch wykonany przez przeciwnika (komputer) oraz aktualny stan planszy.

	KOLUMNY									
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
KOLUMNY										
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[B]	[]	[]	[]	[]	[]	3	R
4	[]	[W]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
0	1	2	3	4	5	6	7			

Aktualnie przy ruchu: 1
Podaj współrzędne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyć:

Rys. 7. Sposób wykonania ruchu przez przeciwnika (komputer)

- 12) Bicie pionem polega na przeskoczeniu pionu przeciwnika znajdującego się na sąsiednim (po przekątnej) polu i zajęcie pola znajdującego się tuż za nim (po przekątnej). W tym celu należy podać współrzędne pola, na którym znajduje się pion do zbitia. Zbite piony są usuwane z planszy po zakończeniu ruchu.

	KOLUMNY									
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
KOLUMNY										
0	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[B]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[W]	[]	[]	[]	[]	4	S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
0	1	2	3	4	5	6	7			

Aktualnie przy ruchu: 1
Podaj współrzędne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyć:
4 3
from: 4 3 to: 3 4
from: 4 3 to: 3 2
Podaj gdzie chcesz się ruszyć
3 2

	KOLUMNY									
0	[]	[B]	[]	[B]	[]	[B]	[]	[B]	0	W
KOLUMNY										
0	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[W]	[]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[]	4	S
5	[W]	[]	[]	[]	[W]	[]	[W]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
0	1	2	3	4	5	6	7			

Rys. 8. Prawidłowy sposób wykonywania ruchu (bicja)

- 13) Bicia nie są obowiązkowe.

- 14) Bicia do tyłu są niedozwolone.

	KOLUMNY									
	0	1	2	3	4	5	6	7	0	W
0	[]	[]	[]	[]	[]	[]	[]	[B]	0	W
1	[]	[]	[W]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[W]	4	S
5	[W]	[]	[Q_W]	[]	[]	[]	[]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
1 2
from: 1 2 to: 0 3
from: 1 2 to: 0 1
Podaj gdzie chcesz sie ruszyc
3 0
Podaj gdzie chcesz sie ruszyc

Rys. 9. Nieprawidłowy sposób wykonywania ruchu (bicia)

15) Dozwolone są bicia wielokrotne. W jednym ruchu można wykonać więcej niż jedno bicie tym samym pionem, przeskakując przez kolejne piony przeciwnika. W celu wykonania wielokrotnego bicia należy najpierw wykonać bicie pojedyncze. Następnie program poprosi o wybranie ruchu z podanych możliwości poprzez podanie odpowiedniego indeksu.

	KOLUMNY									
	0	1	2	3	4	5	6	7	0	W
0	[]	[]	[]	[B]	[]	[B]	[]	[B]	0	W
1	[B]	[]	[B]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[]	[]	[B]	[]	[B]	2	E
3	[W]	[]	[B]	[]	[B]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[W]	[]	[W]	4	S
5	[W]	[]	[]	[]	[]	[]	[]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
4 5
from: 4 5 to: 3 6
from: 4 5 to: 3 4
Podaj gdzie chcesz sie ruszyc
1 2
Podaj gdzie chcesz sie ruszyc

Rys. 10. Nieprawidłowy sposób wykonywania ruchu (bicia wielokrotnego)

```

    KOLUMNY
  0   1   2   3   4   5   6   7   0   W
0 [ ] [ ] [ ] [B] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [B] [ ] [B] [ ] [B] [ ] 1 I
2 [ ] [B] [ ] [ ] [ ] [B] [ ] [B] 2 E
3 [W] [ ] [B] [ ] [B] [ ] [ ] [ ] 3 R
4 [ ] [ ] [ ] [ ] [ ] [W] [ ] [W] 4 S
5 [W] [ ] [ ] [ ] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] [W] 6 E
7 [W] [ ] [W] [ ] [W] [ ] [W] [ ] 7
  0   1   2   3   4   5   6   7
Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3 ), który chcesz ruszyc:
4 5
from: 4 5 to: 3 6
from: 4 5 to: 3 4
Podaj gdzie chcesz sie ruszyc
3 4
Wybierz mozliwy ruch z podanych:
from: 2 3 to: 1 2
Podaj index ruchu
0

    KOLUMNY
  0   1   2   3   4   5   6   7   0   W
0 [ ] [Q_W] [ ] [B] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [ ] [ ] [B] [ ] [B] [ ] 1 I
2 [ ] [B] [ ] [ ] [ ] [B] [ ] [B] 2 E
3 [W] [ ] [B] [ ] [ ] [ ] [ ] [ ] 3 R
4 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [W] 4 S
5 [W] [ ] [ ] [ ] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] [W] 6 E
7 [W] [ ] [W] [ ] [W] [ ] [W] [ ] 7
  0   1   2   3   4   5   6   7
from: 0 3 to: 1 2

```

Rys. 11.

Prawidłowy sposób wykonania ruchu (bicia wielokrotnego)

16) Podczas bicia można wielokrotnie przechodzić przez to samo puste pole, ale nie można przeskakiwać więcej niż raz przez ten sam pion.

17) Nie można bić swoich pionów.

```

    KOLUMNY
  0   1   2   3   4   5   6   7   0   W
0 [ ] [B] [ ] [B] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [B] [ ] [B] [ ] [B] [ ] 1 I
2 [ ] [ ] [ ] [B] [ ] [B] [ ] [B] 2 E
3 [ ] [ ] [B] [ ] [ ] [ ] [ ] [ ] 3 R
4 [ ] [W] [ ] [ ] [ ] [ ] [ ] [ ] 4 S
5 [W] [ ] [ ] [ ] [W] [ ] [W] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] [W] 6 E
7 [W] [ ] [W] [ ] [W] [ ] [W] [ ] 7
  0   1   2   3   4   5   6   7
Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3 ), który chcesz ruszyc:
6 3
from: 6 3 to: 5 2
Podaj gdzie chcesz sie ruszyc
5 4
Podaj gdzie chcesz sie ruszyc

```

Rys. 12. Nieprawidłowy sposób wykonania ruchu (bicia)

18) Pion, który dojdzie do ostatniego rzędu planszy, staje się damką.

KOLUMNY										
0	1	2	3	4	5	6	7	[B]	0	W
0	[]	[]	[]	[]	[]	[]	[]	[B]	0	W
1	[]	[]	[W]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[W]	4	S
5	[W]	[]	[Q_W]	[]	[]	[]	[]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Aktualnie przy ruchu: 1
 Podaj współzadne pionka: wiersz kolumna (np. 3 3), który chcesz ruszyc:
 1 2
 from: 1 2 to: 0 3
 from: 1 2 to: 0 1
 Podaj gdzie chcesz sie ruszyc
 3 0
 Podaj gdzie chcesz sie ruszyc
 0 3

KOLUMNY										
0	1	2	3	4	5	6	7	[B]	0	W
0	[]	[]	[]	[Q_W]	[]	[]	[]	[B]	0	W
1	[]	[]	[]	[]	[B]	[]	[B]	[]	1	I
2	[]	[B]	[]	[B]	[]	[B]	[]	[B]	2	E
3	[]	[]	[]	[]	[]	[]	[]	[]	3	R
4	[]	[]	[]	[]	[]	[]	[]	[W]	4	S
5	[W]	[]	[Q_W]	[]	[]	[]	[]	[]	5	Z
6	[]	[W]	[]	[W]	[]	[W]	[]	[W]	6	E
7	[W]	[]	[W]	[]	[W]	[]	[W]	[]	7	
	0	1	2	3	4	5	6	7		

Rys. 13. Prawidłowy sposób wykonywania ruchu

- 19) Jeżeli pion dojdzie do ostatniego rzędu planszy w wyniku bicia i w tym samym ruchu wykona kolejne bicie (do tyłu), nie staje się damką.
- 20) Kiedy pion staje się damką, kolej ruchu przypada na przeciwnika.
- 21) Damki mogą poruszać się o jedno pole do przodu lub do tyłu po przekątnej.

```

    KOLUMNY
  0   1   2   3   4   5   6   7
0 [ ] [ ] [ ] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [ ] [ ] [ ] [B] [ ] 1 I
2 [ ] [B] [ ] [B] [ ] [B] [ ] 2 E
3 [W] [ ] [ ] [ ] [ ] [ ] [ ] 3 R
4 [ ] [Q_W] [ ] [ ] [ ] [ ] [ ] 4 S
5 [W] [ ] [ ] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] 6 E
7 [W] [ ] [W] [ ] [W] [W] [ ] 7
  0   1   2   3   4   5   6   7
Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3 ), który chcesz ruszyc:
4 1
from: 4 1 to: 5 2
from: 4 1 to: 3 2
Podaj gdzie chcesz się ruszyc
5 2

    KOLUMNY
  0   1   2   3   4   5   6   7
0 [ ] [ ] [ ] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [ ] [ ] [ ] [B] [ ] 1 I
2 [ ] [B] [ ] [B] [ ] [B] [ ] 2 E
3 [W] [ ] [ ] [ ] [ ] [ ] [ ] 3 R
4 [ ] [ ] [ ] [ ] [ ] [ ] [ ] 4 S
5 [W] [ ] [Q_W] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] 6 E
7 [W] [ ] [W] [ ] [W] [W] [ ] 7
  0   1   2   3   4   5   6   7
from: 0 5 to: 1 4

```

Rys. 14. Prawidłowy sposób wykonywania ruchu

22) Damki mogą być piony przeciwnika do przodu lub do tyłu.

```

    KOLUMNY
  0   1   2   3   4   5   6   7
0 [ ] [Q_W] [ ] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [B] [ ] [B] [ ] [B] 1 I
2 [ ] [B] [ ] [ ] [B] [ ] [B] 2 E
3 [W] [ ] [B] [ ] [ ] [ ] [ ] 3 R
4 [ ] [ ] [ ] [ ] [ ] [ ] [ ] 4 S
5 [W] [ ] [ ] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] 6 E
7 [W] [ ] [W] [ ] [W] [W] [ ] 7
  0   1   2   3   4   5   6   7
Aktualnie przy ruchu: 1
Podaj współzadne pionka: wiersz kolumna (np. 3 3 ), który chcesz ruszyc:
0 1
from: 0 1 to: 1 2
Podaj gdzie chcesz się ruszyc
1 2
Wybierz możliwy ruch z podanych:
from: 2 3 to: 3 2
Podaj index ruchu
0

    KOLUMNY
  0   1   2   3   4   5   6   7
0 [ ] [ ] [ ] [ ] [B] [ ] [B] 0 W
1 [B] [ ] [ ] [ ] [B] [ ] [B] 1 I
2 [ ] [B] [ ] [ ] [B] [ ] [B] 2 E
3 [W] [ ] [ ] [ ] [ ] [ ] [ ] 3 R
4 [ ] [Q_W] [ ] [ ] [ ] [ ] [ ] 4 S
5 [W] [ ] [ ] [ ] [ ] [ ] [ ] 5 Z
6 [ ] [W] [ ] [W] [ ] [W] [ ] 6 E
7 [W] [ ] [W] [ ] [W] [W] [ ] 7
  0   1   2   3   4   5   6   7
from: 1 4 to: 2 3

```

Rys. 15. Prawidłowy sposób wykonywania ruchu

23) Celem gry jest zbitie wszystkich pionów przeciwnika lub zablokowanie wszystkich pionów pozostających za planszy. [4]

4. Wnioski

Przedstawiony obcięty algorytm minimaks jest algorytmem, który może być z dobrym skutkiem zastosowany w praktyce. Z odpowiednio starannie zaprojektowaną funkcją heurystyczną może w wielu grach skutecznie konkurować przynajmniej z średniej klasy graczami-ludźmi. W grze w warcaby funkcja heurystyczna jest obliczana z uwzględnieniem liczby poszczególnych figur w posiadaniu gracza i przeciwnika. Czasem uwzględniane jest także ich położenie czy też występowania specyficznych układów sąsiedztwa figur.

W najbardziej skutecznych programach grających w gry stosowane są niezwykle wyrafinowane funkcje heurystyczne, oparte na najwyższej klasy wiedzy eksperckiej i szeroko zakrojonych eksperymentach. Szczególnie obiecujący jest także pomysł, aby funkcja heurystyczna mogła podlegać modyfikacjom na podstawie rozgrywanych partii, w sposób poprawiający dokładność, z jaką pozwala ona przewidywać szanse wygranej w nierozstrzygniętych sytuacjach. Jest to najbardziej naturalna technika wyposażenia programów grających w gry w zdolność do uczenia się.

W konkretnej partii gry z przeciwnikiem, który nie zawsze wybiera optymalne dla siebie ruchy minimaks, może istnieć ruch prowadzący do lepszego wyniku partii niż ruch optymalny minimaks, lecz w sytuacji, gdy strategia wyboru ruchu stosowana przez przeciwnika nie jest znana, nie można wykorzystać takich sytuacji. Jednocześnie można się spodziewać, że w większości sytuacji nieoptymalne zachowanie się przeciwnika raczej ułatwi niż utrudni sytuację gracza, a zatem stosowanie zasady minimaks pozostaje uzasadnione także przy niskim poziomie kompetencji przeciwnika.

5. Literatura

- [1] <http://wyborcza.pl/1,75400,4326808.html> (data dostępu: 28.05.2019)
- [2] https://pl.wikipedia.org/wiki/Teoria_gier (data dostępu: 28.05.2019)
- [3] http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modus%82_8_-_Gry_dwuosobowe (data dostępu: 28.05.2019)
- [4] <http://archiwum.warcaby.pl/kodeks-warcabowy/134-rozdzial-i-oficjalne-reguly-gry-w-warcaby> (data dostępu: 28.05.2019)