

Sztuczna inteligencja i inżynieria wiedzy

Lista 2

Gabriel Urbaniak 260428

2 maja 2023

1 Problem

1.1 Reversi

Reversi to gra:

- dwuosobowa,
- o sumie zerowej,
- niekooperacyjna,
- całkowicie deterministyczna,
- w której każdy gracz ma pełną informację o stanie gry,
- o skończonym czasie rozgrywki.

Zadaniem gracza w Reversi jest zwycięstwo przez stawianie pionków na planszy w taki sposób, by przejąć pionki przeciwnika - na koniec gry należy posiadać najwięcej pionków, by wygrać.

Definicja 1 (Gracz) Gracz $\mathbf{G} \in \{B, C\}$ to gracz grający w grę Reversi. Może wykonać ruch po ruchu przeciwnika lub jako pierwszy, gdy $\mathbf{G} = B$

Definicja 2 (Plansza) Plansza to macierz $\mathbf{P} = (p_{ij})_{1 \leq i \leq 8, 1 \leq j \leq 8}$, gdzie p to pole na tej planszy. Pole może mieć trzy różne stany: $\mathbf{S} = \{0, B, C\}$, $p_{ij} \in \mathbf{S}$.

Definicja 3 (Ruch) Gracz może się ruszyć przez postawienie swojego pionka na dowolne z wolnych pól $p_{ij} = 0$ zmieniając jego symbol na B lub C zależnie od symbolu gracza dopóki jest w stanie przejąć pionki przeciwnika. Jeśli nie jest w stanie przejąć ani jednym ruchem pionka przeciwnika, to ruch gracza jest pomijany.

Definicja 4 (Przejęcie) Przy postawieniu pionka jednego z graczy G na pole p_{ij} należy sprawdzić, czy sąsiadujące pola $p_{i\pm 1j\pm 1}$ należą do przeciwnika. Jeśli tak, to wszystkie pionki przeciwnika w linii prostej w pionie, poziomie lub przekątnej macierzy przecinającą pole p_{ij} w kierunku tego pionka przeciwnika zostają przejęte przez gracza G , czyli zmienione na jego symbol, pod warunkiem, że linia ta kończy się również jego pionkiem.

Definicja 5 (Koniec gry) Gra się kończy, gdy żaden z graczy G nie może wykonać ruchu. Zliczane są wszystkie pola planszy P według ich stanu S . Jeśli liczba czarnych pionków i białych jest taka sama, żaden z graczy nie wygrywa. W przeciwnym przypadku wygrywa gracz, który posiada na planszy więcej pionków od drugiego.

Definicja 6 (Stan gry) Stan gry to $R_s = (P, G)$, gdzie P to plansza, a G to gracz, który miałby wykonywać teraz ruch.

Definicja 7 (Stan początkowy) Początkowym stanem gry jest:

$$R_s' = (P', B)$$

$$P' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B & C & 0 & 0 & 0 \\ 0 & 0 & 0 & C & B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

1.2 Metoda

Problem można było rozwiązać przez sprawdzenie każdego możliwego ruchu jaki istnieje, jednak wymagałoby to ogromnej mocy obliczeniowej. Z tego powodu najlepszy możliwy ruch w danej sytuacji będzie znajdowany przez częściowe rozwinięcie drzewa gry. Aby ocenić grę, która jeszcze się nie zakończyła należy stworzyć **funkcję oceniającą** która mogłaby wartością numeryczną ocenić dobroć sytuacji dla algorytmu przeszukującego **drzewo decyzyjne**.

Definicja 8 (Drzewo decyzyjne) Drzewo decyzyjne to $N = (C, S, R_s)$, $C = \{N_1, N_2, \dots, N_n\} \cup \emptyset$, $S \in \mathbb{R}$, gdzie N jest węzłem, a S wynikiem funkcji oceniającej dla danego węzła.

Definicja 9 (Funkcja oceniająca (heurystyczna))

$$f = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n$$

f to funkcja oceniająca, gdzie x_n to pewna miara stanu planszy, a w_n to waga odpowiedniej miary stanu planszy.

Algorytmem, który będzie wykorzystywał owe dane będzie **Minimax** oraz jego modyfikacja - **Alfa-beta cięcie**, które pozwoli na omijanie węzłów które już nie ma sensu przeszukiwać, by oszczędzić trochę mocy obliczeniowej w celu poprawy np. głębokości drzewa bądź czasu wykonania algorytmu.

2 Implementacja

Zasady gry zaimplementował Dr inż. Piotr Syga[1].

Implementacja pozwala na zmierzenie się ze sobą dwóch dowolnych graczy, gdzie graczami mogą być Minimax, Alfa-beta cięcie bądź człowiek. Wybranymi miarami stanu planszy zostały:

- Liczba posiadanych pionków
- Elastyczność - liczba dostępnych ruchów
- Blokowanie - ujemna liczba dostępnych ruchów przeciwnika (kara za to, że ma ruchy)
- Liczba pionków przeciwnika sąsiadująca z pionkami algorytmu (potencjalne okazje do przejęcia)
- Liczba pionków przeciwnika sąsiadujących z pustymi polami. (potencjalne okazje do przejęcia)

W programie można stworzyć własne tabele wag (strategii), jednak zostało umieszczone w nim kilka przykładowych:

nazwa	liczba pionków	elastyczność	blokowanie	sąsiedzi pionków przeciwnika	okazje
aggressive	4.0	2.0	0.5	1.5	2.0
controlling	1.0	5.0	5.0	1.0	1.0
balanced	1.0	2.0	2.0	1.0	1.0

Kod znajduje się na tej stronie.

3 Rezultaty

Podstawowymi mierzalnymi parametrami do porównania są czas wykonania oraz ilość odwiedzonych węzłów.

Przedstawione zostaną dane przykładowej partii między algorytmem Minimax a jego zmodyfikowanej wersji z Alfa-beta cięciem.

Parametrami obu algorytmów są:

Głębokość: 4 Strategia: balanced

Minimax	czas	węzły	Alfa-beta	czas	węzły
(2, 4)	0.100894	317	(4, 5)	0.063711	198
(5, 2)	0.267528	847	(2, 3)	0.112144	344
(2, 5)	0.82	2573	(2, 2)	0.246547	742
(4, 2)	1.18808	3736	(5, 3)	0.215184	654
(3, 2)	2.10616	6834	(4, 1)	0.635999	1949
(3, 1)	4.601079	15131	(1, 5)	0.680569	2200
(5, 4)	5.337107	18047	(3, 0)	0.761715	2505
(1, 3)	4.264432	14603	(5, 5)	0.673579	2231
(3, 5)	6.348387	22341	(6, 5)	0.903315	3040
(6, 4)	9.784564	35112	(1, 2)	0.829852	2851
(5, 1)	11.259568	40868	(7, 4)	1.229304	4242
(5, 6)	7.426985	27506	(6, 6)	2.254521	8138
(7, 7)	11.504372	43884	(6, 2)	2.071016	7393
(4, 0)	13.258387	52211	(6, 3)	1.585612	5695
(1, 4)	8.666449	35624	(2, 0)	0.994522	3671
(7, 6)	11.038945	47129	(1, 1)	1.226644	4705
(4, 6)	8.912769	40016	(3, 7)	1.217893	4968
(2, 1)	8.762945	39649	(7, 5)	1.064265	4432
(4, 7)	7.298406	34770	(0, 4)	0.331787	1397
(0, 0)	4.024789	20193	(1, 0)	0.163105	719
(7, 3)	2.349871	11774	(7, 2)	0.178921	858
(6, 1)	1.40726	7741	(0, 2)	0.164144	814
(0, 3)	0.595515	3571	(6, 0)	0.131961	689
(5, 0)	0.540275	3338	(3, 6)	0.077873	442
(2, 7)	0.277857	1861	(2, 6)	0.061833	375
(0, 5)	0.116634	838	(0, 6)	0.019758	130
(0, 1)	0.040778	332	(1, 6)	0.007116	52
(7, 0)	0.011538	121	(5, 7)	0.00296	27
(0, 7)	0.001524	30	(1, 7)	0.000286	5
(6, 7)	0.000165	4	(7, 1)	6.4e-05	2

Zwycięstwo gracza 1. Tur: 60

Programy odwiedzają najwięcej węzłów w środkowej fazie gry, zaś na początku i końcu jest ich o wiele mniej. Alfa-beta cięcie jest zdecydowanie szybszy - odwiedza o wiele mniej węzłów. Nie były one sprawdzane, ponieważ i tak nie miałyby to sensu ze względu na brak poprawy wyniku. Zmiana rozpoczynającego algorytmu nie ma wpływu na partię - algorytmy wykonują te same ruchy. Ze względu na brak różnicy w "myśleniu" algorytmów a zdecydowanie szybszym czasem wykonania od teraz wszystkie testy będą wykonywane Alfa-beta cięciem.

Porównanie zmierzających się ze sobą strategii:

Głębokość: 3

Gracz 1.	Gracz 2.	Zwycięstwo
controlling	aggressive	2
aggressive	balanced	2
controlling	balanced	1
aggressive	controlling	2
balanced	aggressive	2
balanced	controlling	2

W przypadku pojedynku balans a kontrola wygrywa zawsze kontrola. Mecze z agresywnym zawodnikiem bywają chaotyczne - ciężko stwierdzić kto wygra.

Porównanie głębokości i wygranych:

Strategia	Gracz 1.	Gracz 2.	Zwycięstwo
controlling	2	4	2
aggressive	2	4	2
balanced	2	4	2
controlling	4	2	1
aggressive	4	2	1
balanced	4	2	1

Zdecydowanie wyższe prawdopodobieństwo na wygraną ma algorytm z większą głębokością.

4 Podsumowanie

Alfa-beta cięcie jest zdecydowanie szybszym algorytmem. Nie traci on również na jakości rozwiązania.

Bardzo ważnym jest znalezienie dobrych metryk sytuacji stanu gry, a następnie dobranie odpowiedniego wektora wag. Jeśli to nie zwycięstwo jest

celem a rekreacja, to można stworzyć różne, ciekawe strategie przeciwko którym można staczać ciekawe rozgrywki. Oprócz tego można wprowadzić "anty-strategie", dzięki którym można obniżyć poziom kompetencji algorytmu i sprawić, że rozgrzywka będzie mniej wymagająca dla nowych graczy.

Bibliografia

- [1] Dr inż. Piotr Syga. *Mechanika Reversi*. URL: <https://syga.kft.pwr.edu.pl/courses/siiw/reversi.py>. (dostępne 1.05.2023).