

Sztuczna inteligencja i inżynieria wiedzy

Sprawozdanie z zadania trzeciego laboratorium

„Algorytmy rozwiązywania gier o sumie zerowej”

Prowadzący laboratorium:
dr. inż. Paweł Myszkowski

Przygotował:
Aleksander Poławski

1. Wstęp

Celem zadania trzeciego była implementacja popularnej gry logicznej „Connect4” z możliwością rozgrywek człowiek przeciw człowiekowi, człowiek przeciw AI, AI przeciw AI oraz na jej przykładzie implementacja i badanie skuteczności algorytmów min-max i alfa-beta.

a) Przeanalizowano główne pojęcia związane z grami:

Gra – według teorii gier von Neumann’a i Morgenstern’a składa się z zestawu reguł określających możliwości postępowania jednostek znajdujących się w sytuacji określonej mianem konfliktu interesów, w której każda z jednostek stara się maksymalizować swój własny zysk i jednocześnie minimalizować zysk pozostałych graczy.

Reguły gry – określają ilość informacji dostępnych każdemu z graczy oraz wysokości wygranych i przegranych.

Strategia gry – kompletny zbiór zasad, które determinują posunięcie gracza, wybierane dla sytuacji podczas gry.

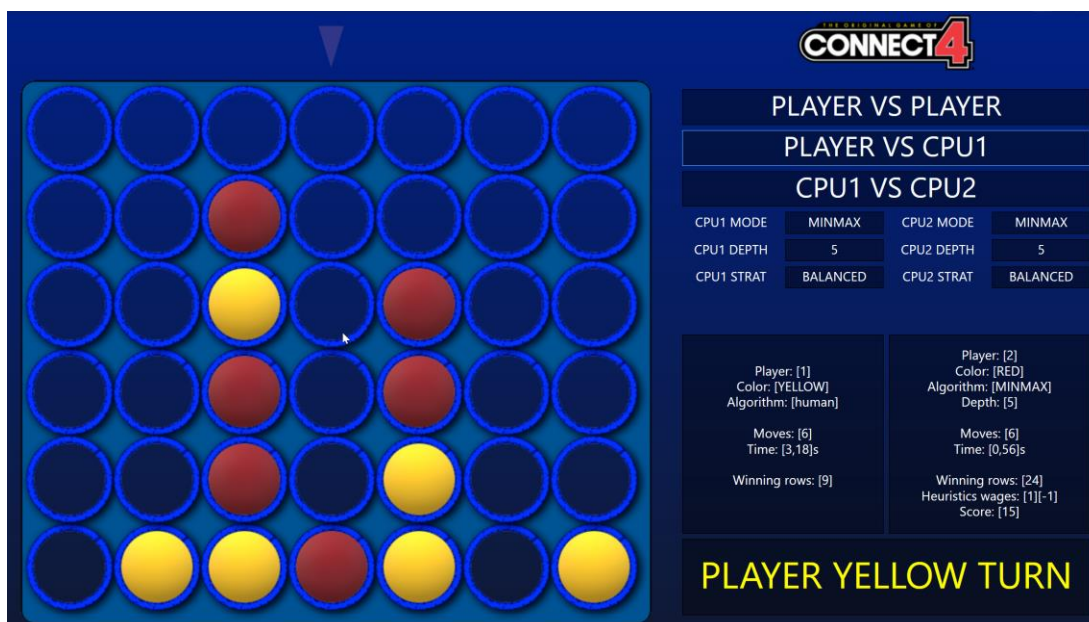
b) Przeanalizowano grę oraz reguły Connect4:

Connect4 jest popularną grą przypominającą najprostsze gry strategiczne, planszowe jak np. Kółko i Krzyżyk. Plansza gry składa się z siedmiu kolumn i sześciu rzędów w których można umieścić żetony. Celem gry jest ułożenie ciągu czterech żetonów w pionie, poziomie lub na ukos. Ruch gracza polega na wyborze kolumny (po jej wybraniu żeton umieszcza się na pierwszej wolnej pozycji od dołu kolumny). W fizycznej wersji gry plansza ustawiana jest pionowo, a gracze wrzucają żetony od góry. Ruchy graczy wykonywane są naprzemiennie.

Gra posiada stosunkowo małą przestrzeń stanów (rzędu 10^{15}) oraz małą złożoność drzewa gry (10^{21}). Ciekawostką jest fakt, że została ona rozwiązana i gracz rozpoczynający w optymalnej rozgrywce jest zawsze graczem wygrywającym (algorytm PERFECT).

2. Szczegóły implementacji

a) Program zrealizowano jako aplikację desktopową WPF (Windows Presentation Foundation) w języku C#. Poniżej przedstawiono przykładowy zrzut ekranu działania programu:



b) Ustawienia umożliwiają zmianę algorytmu gracza komputerowego na algorytm Minmax, Alpha-Beta lub Random. W przypadku algorytmów Minmax lub Alpha-Beta istnieje możliwość zmiany głębokości przeszukiwania oraz wyboru strategii.

c) Funkcję oceny planszy zrealizowano jako kombinację liniową dwóch heurystyk - liczby wygrywanych kombinacji na planszy k1 (dowolny ciąg czterech sąsiadujących slotów, gdzie przynajmniej jeden zajęty jest przez żeton gracza i żaden nie jest zajęty przez przeciwnika) oraz liczby przegrywanych kombinacji k2 (analogicznie).

$$\text{Wynik} = k1 * w1 + k2 * w2,$$

gdzie w1 i w2 to odpowiednie wagi dobierane na podstawie strategii wybranej w ustawieniach:

- „BALANCED”: w1 = 1 oraz w2 = -1
- „PREFER WIN”: w1 = 2 oraz w2 = -1
- „ONLY WIN”: w1 = 1 oraz w2 = 0

d) w przypadku gry AI kontra AI istnieje możliwość doboru algorytmów i ich parametrów dla każdego z osobna

3. Algorytm Alpha-Beta kontra algorytm Minmax

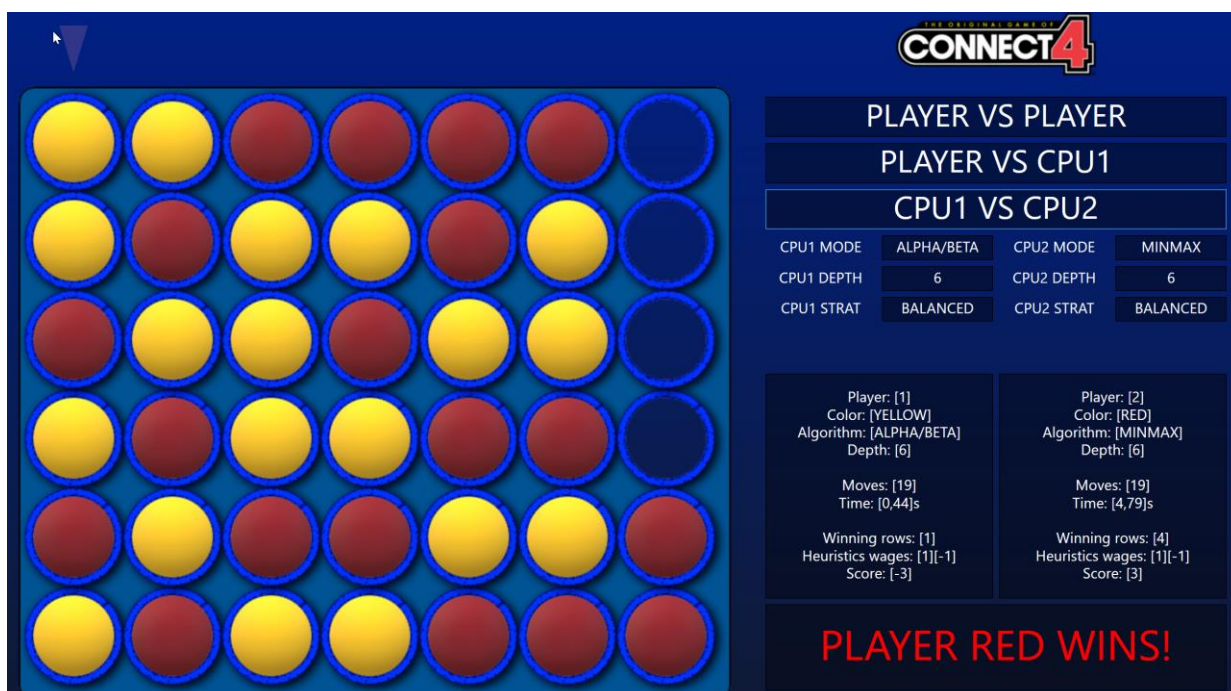
Algorytm Alpha-Beta jest w rzeczywistości usprawnionym algorytmem Minmax. W przypadku kiedy pierwszy ruch nie jest losowy, a ustawienia głębokości i strategii są identyczne, gra potoczy się w ten sam sposób, niezależnie od wybranego algorytmu.

Usprawnienie Alpha-Beta pozwala na wyeliminowanie konieczności przeglądania niektórych gałęzi drzewa, a w związku z tym znacząco skrócić działanie algorytmu.

a) Poniżej przedstawiono stany końcowe gry AI vs AI dla ustawień:

CPU1 – AlphaBeta, Głębokość: 6, strategia: „BALANCED”

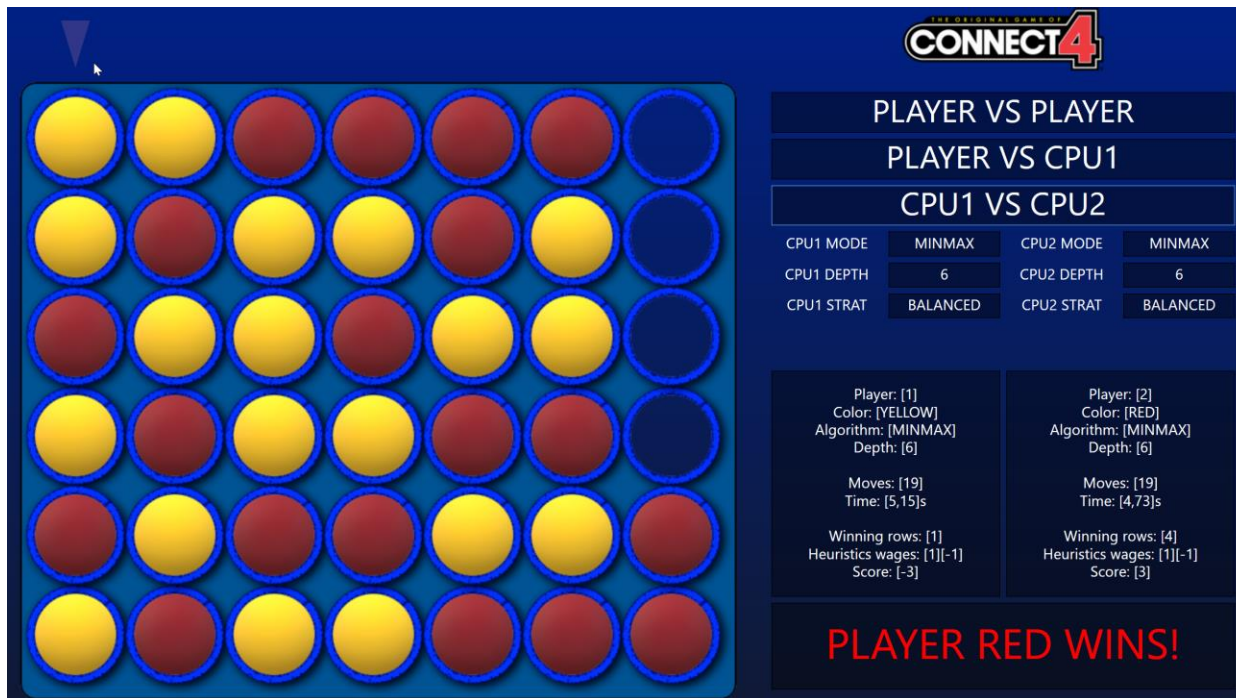
CPU2 – Minmax, Głębokość: 6, strategia: „BALANCED”



Oraz

CPU1 – Minmax, Głębokość: 6, strategia: „BALANCED”

CPU2 – Minmax, Głębokość: 6, strategia: „BALANCED”



Badanie potwierdza, że zachowanie algorytmów Minmax i Alpha-Beta jest takie same w przypadku identycznych ustawień. Zarówno stany końcowe planszy jak i ilość wykonanych ruchów nie zmieniły się po doborze innego algorytmu.

b) Zbadano czasy wykonania pierwszego ruchu dla analogicznych rozgrywek korzystając z dwóch różnych algorytmów. Badanie przeprowadzono powtarzając wykonanie poszczególnych algorytmów 15 razy oraz uśredniając wyniki.

Głębokość przeszukiwania	Minmax – czas wykonania (średnia z 15 uruchomień)	Alpha-Beta – czas wykonania (średnia z 15 uruchomień)
4	0.01s	0.01s
5	0.10s	0.02s
6	0.68s	0.06s
7	4.91s	0.21s
8	31.82s	0.84s
9	>100s	2.70s
10	>>100s	14.40s

Badanie potwierdza wyższość algorytmu Alpha-Beta nad algorytmem Minmax pod względem czasu wykonania.

c) Opierając się na uzyskanych wynikach i poczynionych spostrzeżeniach postanowiono prowadzić dalsze badania korzystając z algorytmu Alpha-Beta.

4. Badanie parametru strategii

Tak jak opisano w szczegółach implementacji:

Funkcję oceny planszy zrealizowano jako kombinację liniową dwóch heurystyk - liczby wygrywanych kombinacji na planszy k1 (dowolny ciąg czterech sąsiadujących slotów, gdzie przynajmniej jeden zajęty jest przez żeton gracza i żaden nie jest zajęty przez przeciwnika) oraz liczby przegrywanych kombinacji k2 (analogicznie).

$$\text{Wynik} = k1 * w1 + k2 * w2,$$

gdzie w1 i w2 to odpowiednie wagi dobierane na podstawie strategii wybranej w ustawieniach:

- „BALANCED”: w1 = 1 oraz w2 = -1
- „PREFER WIN”: w1 = 2 oraz w2 = -1
- „ONLY WIN”: w1 = 1 oraz w2 = 0

Zbadano jakość odpowiednich strategii korzystając z trybu AI kontra AI:

- głębokości przeszukiwań obu algorytmów: 7

Dla każdej kombinacji strategii wykonano po 20 uruchomień i zliczono ilość zwycięstw. Wyniki przedstawiono w tabeli poniżej (w postaci **CPU1 WINS/CPU2 WINS/TIES**):

	CPU2 BALANCED	CPU2 PREFER WIN	CPU2 ONLY WIN
CPU1 BALANCED	5/0/15	10/10/0	12/8/0
CPU1 PREFER WIN	15/5/0	11/9/0	14/6/0
CPU1 ONLY WIN	10/10/0	8/12/0	15/5/0

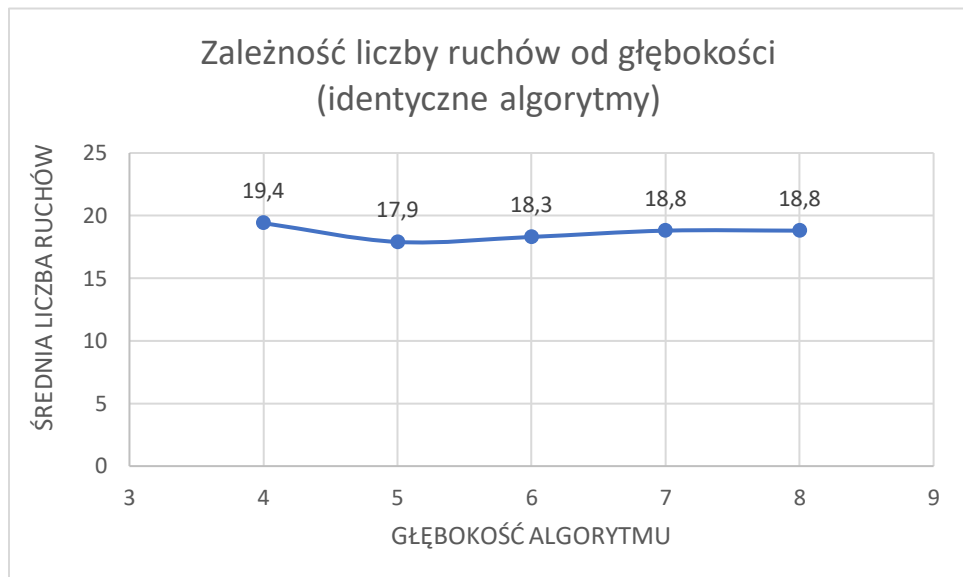
Wnioski:

- najgorszą z heurystyk oceny planszy okazuje się strategia „ONLY WIN”. Ignorowanie liczby przegrywających kombinacji i skupienie się wyłącznie na wygrywających nie jest więc dobrym rozwiązaniem.
- strategia „PREFER WIN” wydaje się nieznacznie przewyższać wyniki uzyskane strategią „BALANCED”.
- okazuje się jednak, że większy wpływ od wyboru strategii w przeprowadzonym badaniu miał wybór pierwszego gracza. Gracz prowadzący znacząco częściej wygrywa rozgrywki niezależnie od ustawionej strategii gry.

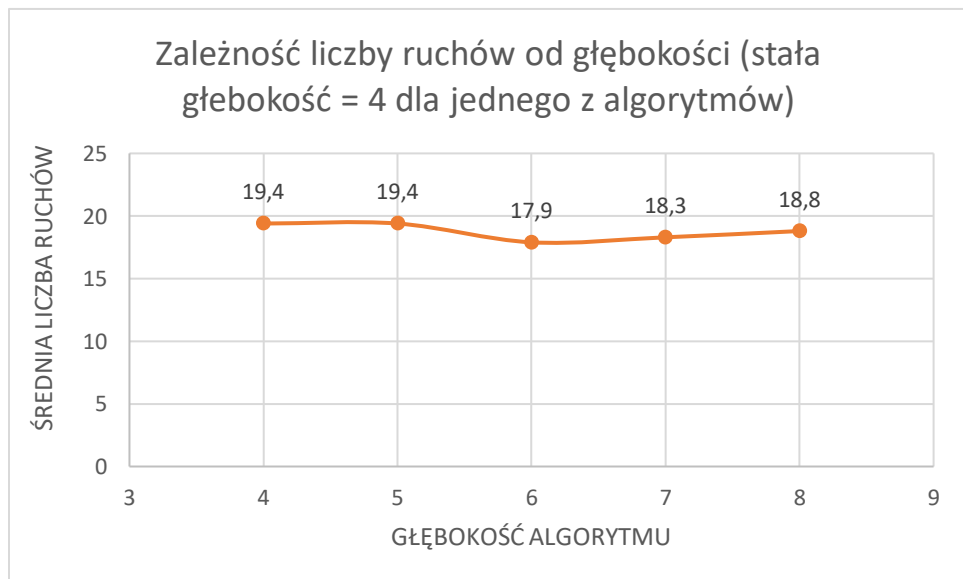
5. Badanie parametru głębokości

Zbadano wpływ głębokości algorytmów na ilość ruchów potrzebnych do wygranej:

a) w przypadku kiedy oba grające ze sobą algorytmy działają na tej samej głębokości (średnia z 20 rozgrywek dla każdej zmiany głębokości):



b) w przypadku kiedy jeden z grających algorytmów działał ze stałą głębokością = 4 (średnia z 20 rozgrywek dla każdej zmiany głębokości):



Nie stwierdzono większej zależności pomiędzy ilością ruchów potrzebnych do zakończenia gry a głębokością przeszukiwania (na przedziale głębokości od 4 do 8). Wynika to z tego, że niezależnie od głębokości, gry kończą się często blisko stanu remisu.

6. Podsumowanie

- a) Już przy tak niskich głębokościach przeszukiwania jak głębokość = 4, zaimplementowane algorytmy nie są trywialne do pokonania, co potwierdza skuteczność algorytmów typu Minmax w rozwiązywaniu gry Connect4.
- b) Algorytm Alpha-Beta jest w każdym przypadku co najmniej tak samo skuteczny jak algorytm Minmax pod względem czasu wykonania i zawsze równie skuteczny pod względem otrzymywanych rezultatów. Istnieje także dalsza możliwość jego optymalizacji poprzez dodanie dodatkowej heurystyki wyboru kolejności sprawdzanych węzłów (w taki sposób, aby szybciej odrzucić niepotrzebne do badania gałęzie drzewa stanów).
- c) Istnieje także możliwość ulepszenia funkcji oceny planszy poprzez kombinację większej ilości heurystyk (oraz ich wag) otrzymując możliwości tworzenia bardziej zróżnicowanych strategii.