

Sztuczna Inteligencja

Algorytmy rozwiązywania gier o sumie zerowej

Weronika Babicz 234797

Spis treści

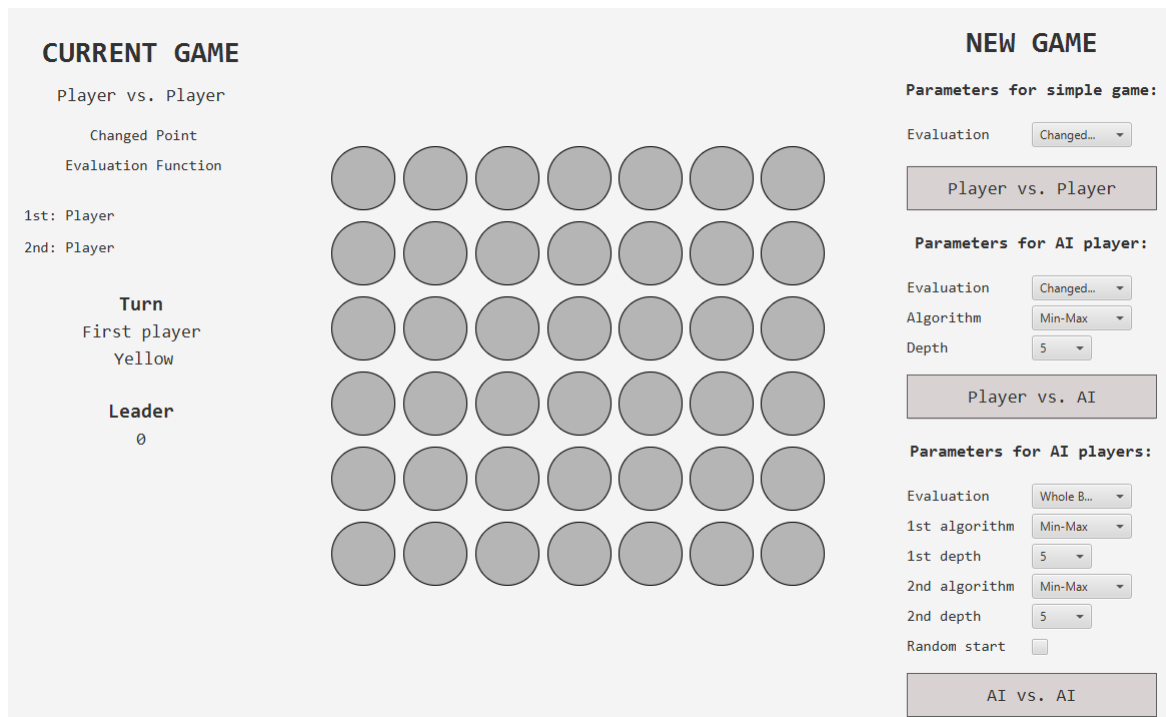
1	Implementacja	3
1.1	Rozgrywka z użyciem sztucznej inteligencji	3
1.1.1	Algorytm Min-Max	3
1.1.2	Algorytm Alfa-Beta	3
1.2	Heurystyki	4
1.2.1	Heurystyka Changed Point	4
1.2.2	Heurystyka Whole Board	4
1.2.3	Heurystyka Changed Point - Blocking	4
1.2.4	Heurystyka Whole Board - Blocking	4
2	Badania	5
2.1	Badania zależności czasu wykonywanych ruchów od głębokości	5
2.1.1	Algorytm Min-Max	5
2.1.2	Algorytm Alfa-Beta	5
2.1.3	Porównanie	6
2.2	Badania wpływu heurystyk na działanie algorytmów	7
2.2.1	Algorytm Min-Max	7
2.2.2	Algorytm Alfa-Beta	8
2.2.3	Porównanie	9
2.3	Badanie wpływu losowości pierwszego ruchu na działanie algorytmów	9
2.4	Badania działania algorytmów przeciwko sobie	10
2.5	Badania czasu wykonywania ruchów dla obu algorytmów	10
2.5.1	Głębokość na poziomie 3	11
2.5.2	Głębokość na poziomie 4	11
2.5.3	Głębokość na poziomie 5	12
2.5.4	Głębokość na poziomie 6	12
2.5.5	Głębokość na poziomie 7	13
3	Podsumowanie	13

1 Implementacja

Zaimplementowano grę Connect4 w trzech trybach gry:

1. Gracz - Gracz
2. Gracz - AI
3. AI - AI

Do projektu dodano interfejs graficzny, aby możliwa była łatwa i wygodna modyfikacja parametrów wybranych rozgrywek.



1.1 Rozgrywka z użyciem sztucznej inteligencji

Dla działania gry z użyciem sztucznej inteligencji zaimplementowano dwa algorytmy odpowiedniego wyboru pola.

1.1.1 Algorytm Min-Max

Algorytm ten zakłada wybranie takiego pola, które zagwarantuje graczowi najlepszy możliwy wynik biorąc pod uwagę zadaną liczbę ruchów w przód.

Algorytm szukając najlepszego rozwiązania dla gracza zawsze zakłada, że przeciwnik wykona najlepszy możliwy ruch dla siebie. Dzięki temu algorytm może podjąć odpowiednią decyzję o wyborze pola.

Min-Max, z racji rekurencyjnej złożoności obliczeniowej, może być wywołany dla maksymalnej głębokości 7 - każda wartość powyżej skutkuje zbyt długim czasem oczekiwania.

Głębokość na poziomie siedmiu oznacza, że sprawdzane będzie dokładnie 7 ruchów naprzód przy założeniu, że każdy z graczy będzie wybierał najlepszy ruch dla siebie. Ostatecznie wybierane pole jest tym, które po sprawdzeniu całego drzewa rozwiązań zagwarantuje najlepsze rozwiązanie.

1.1.2 Algorytm Alfa-Beta

Algorytm Alfa-Beta opiera się na idei algorytmu Min-Max - też sprawdzane i porównywane są wartości planszy dla zadanej głębokości, jednak dodatkową zaletą tego algorytmu jest wykluczanie

poddrzew, które nie zagwarantują już wystarczającego wyniku, aby do nich wchodzić.

Takie rozwiązanie zapewnia cięcia, a co za tym idzie krótszy czas działania programu.

Alfa-Beta wprowadza także nowe parametry - alfę i betę, które oznaczają kolejno najlepszy aktualny wynik dla gracza pierwszego i drugiego.

1.2 Heurystyki

Heurystyki są kluczowym aspektem w działaniu obu algorytmów - na nich opiera się sprawdzanie i porównywanie oceny planszy dla danych etapów gry.

Zdefiniowano cztery heurystyki będące kombinacjami siebie. Każda z nich reprezentuje inną metodę przeliczania planszy - funkcję oceny.

1.2.1 Heurystyka Changed Point

Heurystyka Changed Point zakłada przeliczanie wartości planszy w niewielkim obszarze, a następnie modyfikuje stan rozgrywki (i oceny planszy) sprzed poprzedniego ruchu o tę nową wartość.

Pod uwagę brany jest wiersz, kolumna i dwie przekątne względem punktu, który prezentuje aktualny ruch gracza. W takich zbiorach punktów przyznawane są wyniki odpowiednio przeskalowane w zależności od liczby następujących po sobie punktów należących do aktualnego gracza.

1.2.2 Heurystyka Whole Board

Heurystyka Whole Board z każdą zmianą stanu planszy przelicza jej wynik na nowo, jednak tylko w obszarach, w których są już punkty graczy. Znow pod uwagę brane są wiersze, kolumny i przekątne, jednak są to tylko takie zbiory, które zawierają w sobie chociaż jeden niepusty punkt.

Wyniki za poszczególne ustawienia są przyznawane analogicznie jak w Changed Point.

1.2.3 Heurystyka Changed Point - Blocking

Heurystyka ta pod uwagę bierze zbiory, w których znajduje się ostatni zmodyfikowany punkt, a więc są to dokładnie te same zbiory, jak w przypadku Changed Point.

Zasadniczą różnicą w tej heurystyce jest to, że funkcja oceny planszy bierze pod uwagę konfiguracje, w których gracz jest zablokowany przez przeciwnika i nie ma możliwości wygranej w danej sekwencji. W takim przypadku wyniki są skalowane i odpowiednio pomniejszane.

Jeżeli jednak gracz w danej sekwencji ma tylko swoje punkty, a pozostałe są wolne - jego wynik odpowiednio wzrasta. W tym przypadku brana jest również pod uwagę liczba już postawionych punktów. Jeśli gracz ma ustawione trzy punkty, a pozostały jest wolny - ma większe szanse na wygraną, niż w przypadku gdyby tych punktów miał tylko dwa - a więc w konsekwencji zdobywa lepszy wynik za taki układ.

1.2.4 Heurystyka Whole Board - Blocking

Heurystyka Heurystyka Whole Board - Blocking łączy ze sobą heurystykę Whole Board i Changed Point - Blocking.

Analizowany jest większy obszar planszy oraz brany jest pod uwagę fakt blokowania w funkcji oceny planszy.

2 Badania

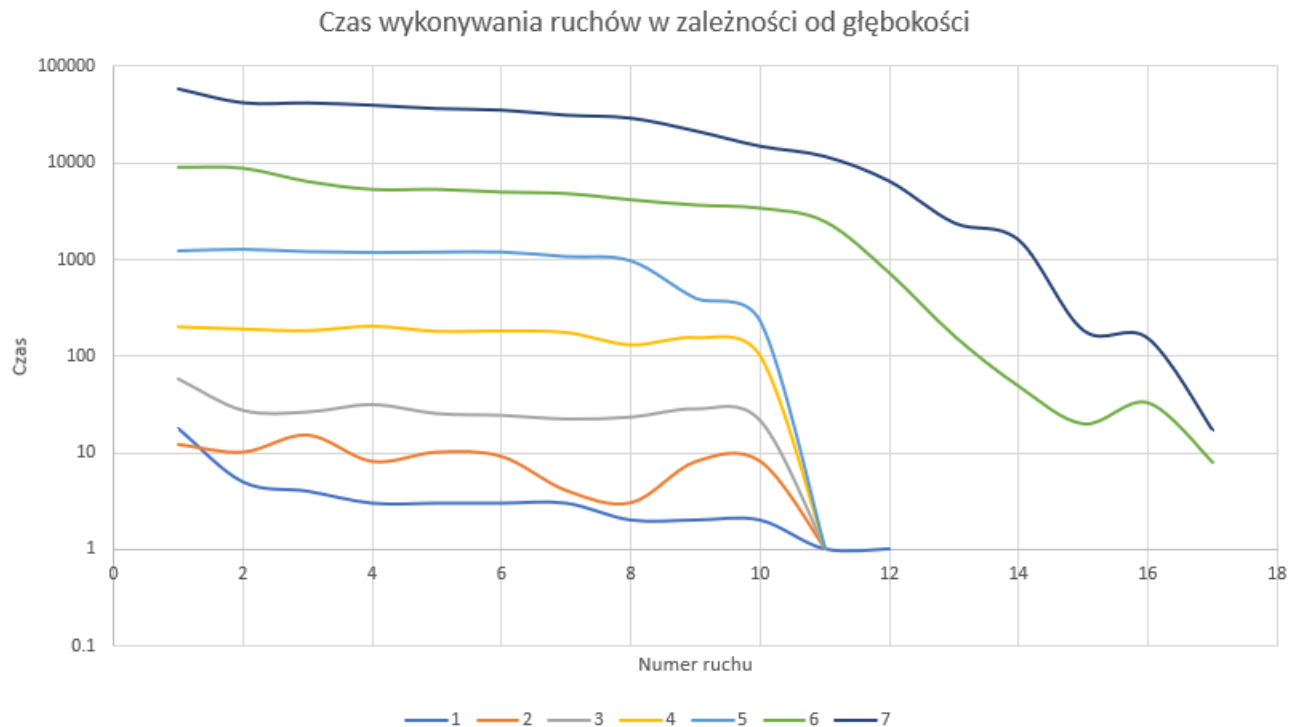
2.1 Badania zależności czasu wykonywanych ruchów od głębokości

2.1.1 Algorytm Min-Max

1 Wykres

Dla zadanych głębokości zbadano czas stawiania kolejnych ruchów przez gracza wykorzystującego algorytm MinMax.

Aby lepiej zobrazować wyniki na wykresie, użyto skali logarytmicznej.



2 Wnioski

Można zauważyć, że czas znalezienia ruchu jest mocno związany z głębokością. Dzieje się tak dlatego, że zwiększając głębokość poszukiwań, zwiększa się liczbę węzłów drzewa, które należy przejrzeć. Liczba sprawdzeń rośnie eksponencjalnie.

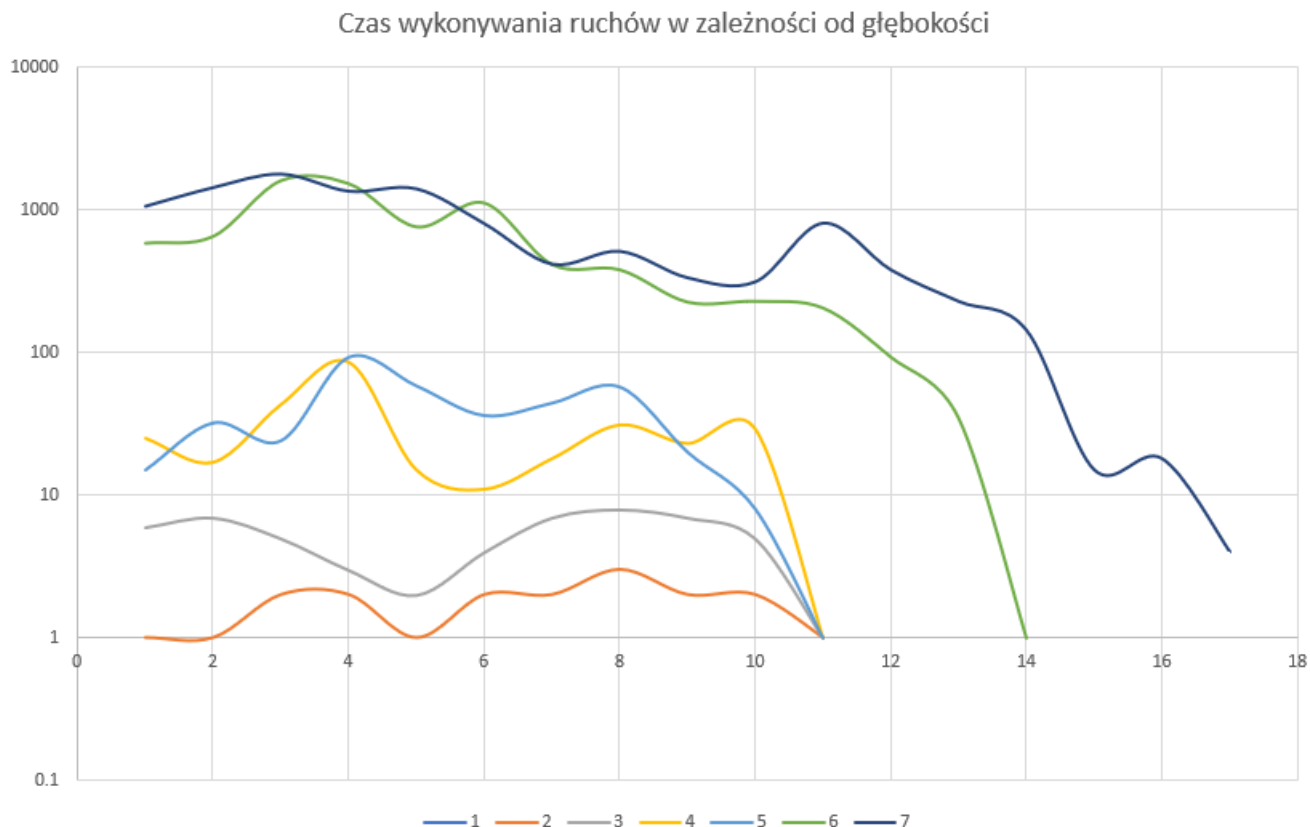
Dodatkowo, zgodnie z oczekiwaniami wraz ze wzrostem głębokości poszukiwań, rośnie liczba ruchów gracza do momentu zwycięstwa (lub zakończenia) gry. Jest to związane z lepszą skutecznością algorytmu dla większych głębokości.

2.1.2 Algorytm Alfa-Beta

1 Wykres

Dla zadanych głębokości zbadano czas stawiania kolejnych ruchów przez gracza wykorzystującego algorytm Alfa-Beta.

Aby lepiej zobrazować wyniki na wykresie, użyto skali logarytmicznej.



2 Wnioski

Można zauważyć, że czas znalezienia ruchu jest mocno związany z głębokością. Dzieje się tak dlatego, że zwiększając głębokość poszukiwań, zwiększa się liczbę węzłów drzewa, które należy przejrzeć. Czas nie rośnie jednak tak szybko jak w przypadku Min-Max ze względu na to, że Alfa-Beta pomija niektóre poddrzewa. Liczba sprawdzeń mimo wszystko jednak rośnie.

Dodatkowo, zgodnie z oczekiwaniami wraz ze wzrostem głębokości poszukiwań, rośnie liczba ruchów gracza do momentu zwycięstwa (lub zakończenia) gry. Jest to związane z lepszą skutecznością algorytmu dla większych głębokości.

2.1.3 Porównanie

Tabela 1: Porównanie średnich czasów wykonywania ruchów oraz liczby ruchów w zależności od algorytmu i głębokości

Algorytm	Miara	Głębokość						
		1	2	3	4	5	6	7
Min-Max	Średni czas	4.18	8.7	28.4	167.9	1013	3499.11	21820.35
	Liczba ruchów	11	10	10	10	10	17	18
Alfa-Beta	Średni czas	0.5	1.8	5.4	29.7	38.6	611.92	611.22
	Liczba ruchów	12	10	10	10	10	13	18

1 Wnioski

Widać znaczną przewagę algorytmu Alfa-Beta w szybkości działania w porównaniu z Min-Max. Jest to spowodowane tym, że Alfa-Beta dokonuje tzw. cięć w miejscach, gdzie przejście

do poddrzewa danego węzła nie przyniesie pożądaných rezultatów. W konsekwencji odrzuca część węzłów i tym samym skraca czas działania operacji.

Dodatkowo, warto wspomnieć o tym, że algorytm Alfa-Beta nie gwarantuje spadku czasowego wraz z każdym kolejnym ruchem, tak, jak to się dzieje w przypadku algorytmu Min-Max. Alfa-Beta, dzięki możliwości wykluczania pewnych poddrzew - nie zapewnia deterministyczności czasowej dla konkretnej głębokości.

2.2 Badania wpływu heurystyk na działanie algorytmów

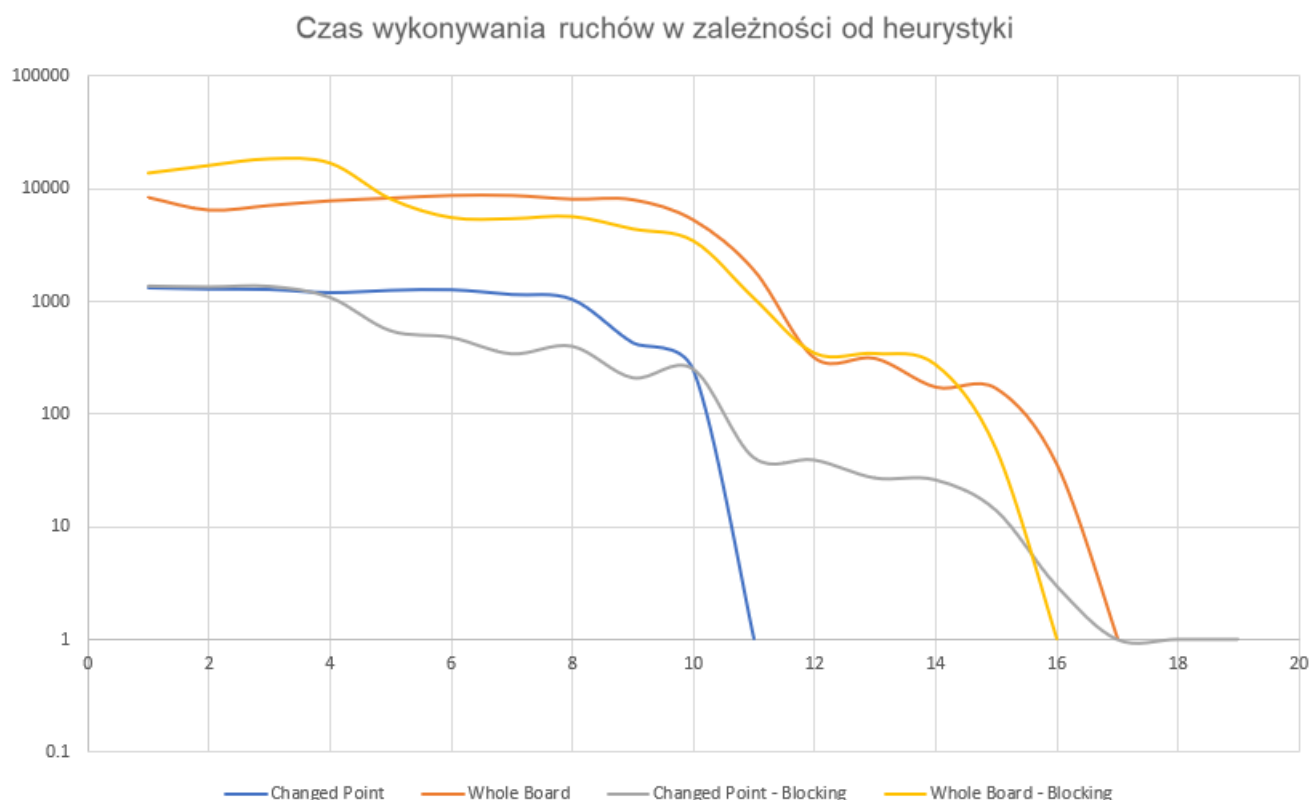
W badaniu przyjęto głębokość optymalną, tzn. taką, która jest wystarczająco dobra, ale też nie zbyt duża, aby nie wydłużać czasu badań. Sprawdzane będą wartości algorytmów dla głębokości 5.

2.2.1 Algorytm Min-Max

1 Wykres

Dla zadanych heurystyk oceny planszy zbadano czas stawiania kolejnych ruchów przez gracza wykorzystującego algorytm Min-Max, a także liczbę ruchów wygrywającego gracza do momentu zwycięstwa (lub zakończenia gry).

Aby lepiej zobrazować wyniki na wykresie, użyto skali logarytmicznej.



2 Wnioski

Czas znajdowania kolejnych ruchów jest silnie związany z wyborem heurystyki. Dla heurystyk z kategorii Whole Board czas jest znacznie dłuższy niż dla pozostałych. Powodem tego jest inny sposób przeliczania wartości planszy; heurystyki Whole Board analizują całą planszę (a konkretniej - te miejsca, w których są już wykonane jakieś ruchy), podczas gdy heurystyki z kategorii

Changed Point analizują zawsze tylko jedno miejsce - to najpóźniej zmodyfikowane na planszy.

Warto także zauważyć, że czasy dla heurystyk Whole Board nie mają tendencji spadkowej przez całą rozgrywkę. Wraz z rozwojem gry, więcej punktów jest na planszy i tym samym jest większy obszar do przeanalizowania, co znacząco wpływa na czas działania programu.

Można także wnioskować, że heurystyki, w których ocena planszy bierze pod uwagę blokowanie sprawiają, że rozgrywka trwa dłużej. Może to świadczyć o tym, że taka heurystyka lepiej oddaje rzeczywistość, jeśli chodzi o odpowiednią ocenę rozgrywki.

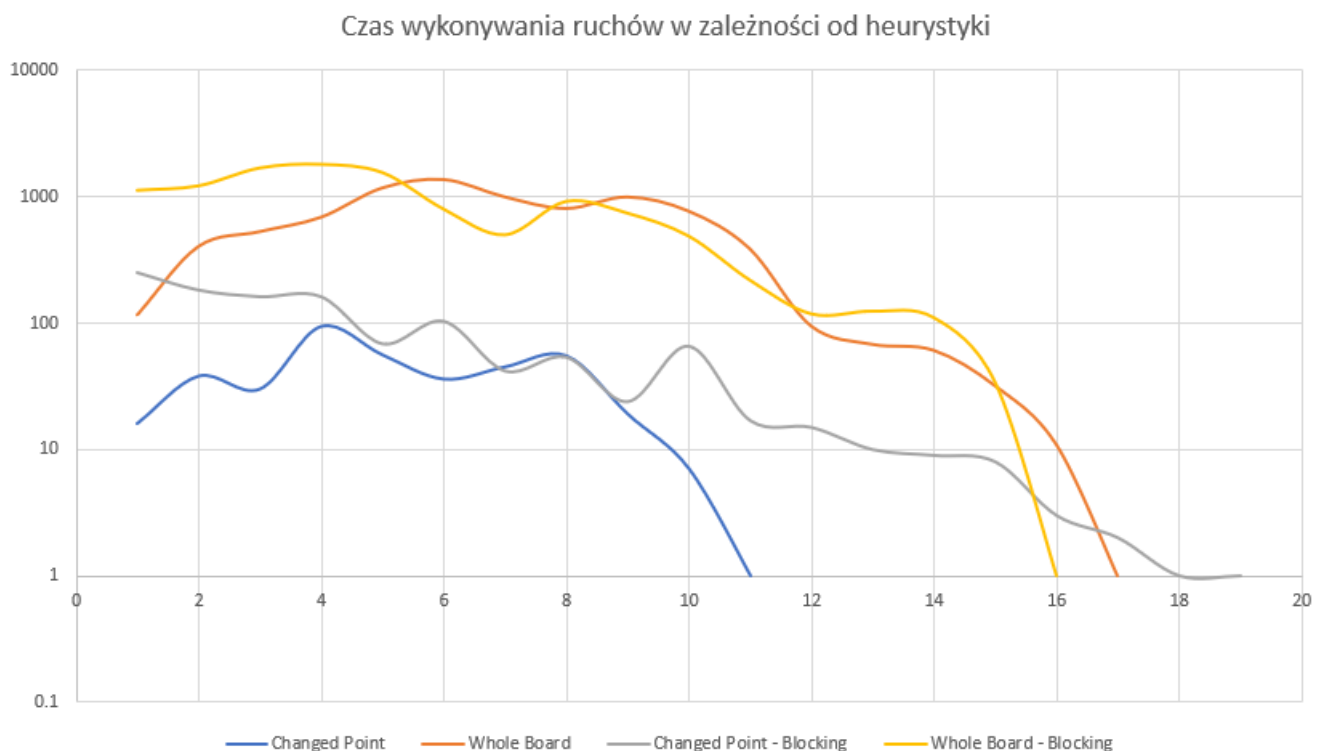
W dodatku, dzięki przeskalowanym punktowaniom za liczbę pustych pól wokół już dokonanych ruchów przez gracza, heurystyka lepiej obrazuje sytuację na planszy niejednoznacznie sugerując lepszą pozycję gracza w przypadku, gdy prawdopodobieństwo wygranej jest coraz większe.

2.2.2 Algorytm Alfa-Beta

1 Wykres

Dla zadanych heurystyk oceny planszy zbadano czas stawiania kolejnych ruchów przez gracza wykorzystującego algorytm Alfa-Beta, a także liczbę ruchów wygrywającego gracza do momentu zwycięstwa (lub zakończenia gry).

Aby lepiej zobrazować wyniki na wykresie, użyto skali logarytmicznej.



2 Wnioski

Czas znajdowania kolejnych ruchów, podobnie jak dla algorytmu Min-Max jest silnie związany z wyborem heurystyki. Dla heurystyk z kategorii Whole Board czas jest znacznie dłuższy niż dla pozostałych. Powodem tego jest inny sposób przeliczania wartości planszy; heurystyki Whole Board analizują całą planszę (a konkretniej - te miejsca, w których są już wykonane jakieś

ruchy), podczas gdy heurystyki z kategorii Changed Point analizują zawsze tylko jedno miejsce - to najpóźniej zmodyfikowane na planszy.

Podobnie jak dla poprzedniego algorytmu - rozgrywki, dla których ocena planszy analizuje blokowanie zdają się działać lepiej. Szczególnie heurystyka Changed Point Blocking, która zapewnia dłuższą rozgrywkę, ale także krótszy czas stawiania ruchów.

2.2.3 Porównanie

Tabela 2: Porównanie średnich czasów wykonywania ruchów oraz liczby ruchów w zależności od algorytmu i heurystyki oceny planszy

Algorytm	Miara	Heurystyka oceny planszy			
		Changed Point	Whole Board	Changed Point Blocking	Whole Board Blocking
Min-Max	Średni czas	1061.4	4925.25	395	6624.13
	Liczba ruchów	10	15	19	16
Alfa-Beta	Średni czas	39.6	533.12	56.61	770.53
	Liczba ruchów	10	15	21	16

1 Wnioski

Poza znaczącą przewagą czasową algorytmu Alfa-Beta można zauważyć także przewagę w skuteczności oceny planszy dla heurystyk z blokowaniem. Dzięki temu, rozgrzywka staje się dłuższa, a jej rozwiązanie ukazuje się w końcowym etapie gry, głównie wtedy, gdy do zapełnienia całej planszy pozostaje bardzo niewiele ruchów.

Dla Changed Point Blocking liczba tych ruchów jest największa, co sugeruje wyrównaną rozgrywkę i fakt, że gracze tak wyrównanie dobierają swoje pola, że gra skutkuje remisem lub wygraną w kilku ostatnich możliwych polach do postawienia (maksymalna liczba ruchów jednego gracza wynosi 21).

2.3 Badanie wpływu losowości pierwszego ruchu na działanie algorytmów

W badaniu tym zestawiono liczbę ruchów do momentu wygrania lub zakończenia rozgrywki dla heurystyki Changed Point - Blocking i dla zadanych głębokości w zależności od losowości wykonania pierwszego ruchu.

Tabela 3: Porównanie liczby ruchów graczy do wygranej lub zakończenia rozgrywki w zależności od losowości wykonania pierwszego ruchu

Algorytm	Losowość	Głębokość						
		1	2	3	4	5	6	7
Min-Max	Tak	7	17	8	14	12	10	5
	Nie	9	21	13	8	21	17	19
Alfa-Beta	Tak	17	20	14	10	16	17	10
	Nie	12	20	13	8	21	17	19

1 Wnioski

Dla algorytmu Min-Max zauważa się wzrost liczby ruchów w przypadku, gdy nie ma losowości wyboru pierwszego ruchu.

Co ciekawe, takie zależności nie zauważa się dla algorytmu Alfa-Beta.

Cieężko stwierdzić jednak istotne znaczenie losowości pierwszego ruchu w skuteczności algorytmu.

2.4 Badania działania algorytmów przeciwko sobie

W tym badaniu zdeterminowano wygrane poszczególnych graczy w danych konfiguracjach parametrów. Funkcją oceny planszy była heurystyka Changed Point - Blocking.

Dodano także pierwszy losowy ruch.

Tabela 4: Porównanie liczby ruchów graczy do wygranej lub zakończenia rozgrywki w zależności od losowości wykonania pierwszego ruchu

		Alpha-Beta		
		3	4	5
Min-Max	3	Alfa-Beta	Alfa-Beta	Alfa-Beta
	4	Min-Max	Alfa-Beta	Alfa-Beta
	5	Min-Max	Alfa-Beta	Alfa-Beta

1 Wnioski

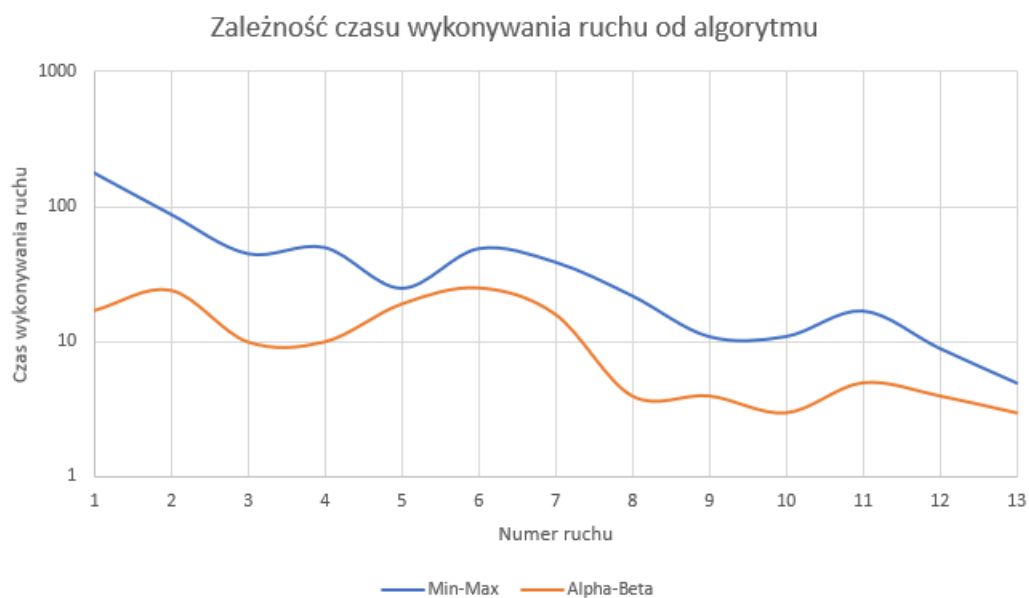
Badanie wykazało przewagę Alfa-Beta nad Min-Maxem. Dla takich samych głębokości zawsze wygrywa Alfa-Beta. Ciekawym wynikiem jest też Min-Max(5) vs. Alfa-Beta(4), w którym więcej sprawdzeń wykonuje Min-Max, jednak to Alfa-Beta zwyciężyła.

Powodem takiego wyniku może być losowość wybrania pierwszego ruchu - najprawdopodobniej wylosowany pierwszy ruch dla Alfa-Bety był znacznie bardziej korzystny niż wylosowany Min-Maxa, co dało temu pierwszemu odpowiednią przewagę.

2.5 Badania czasu wykonywania ruchów dla obu algorytmów

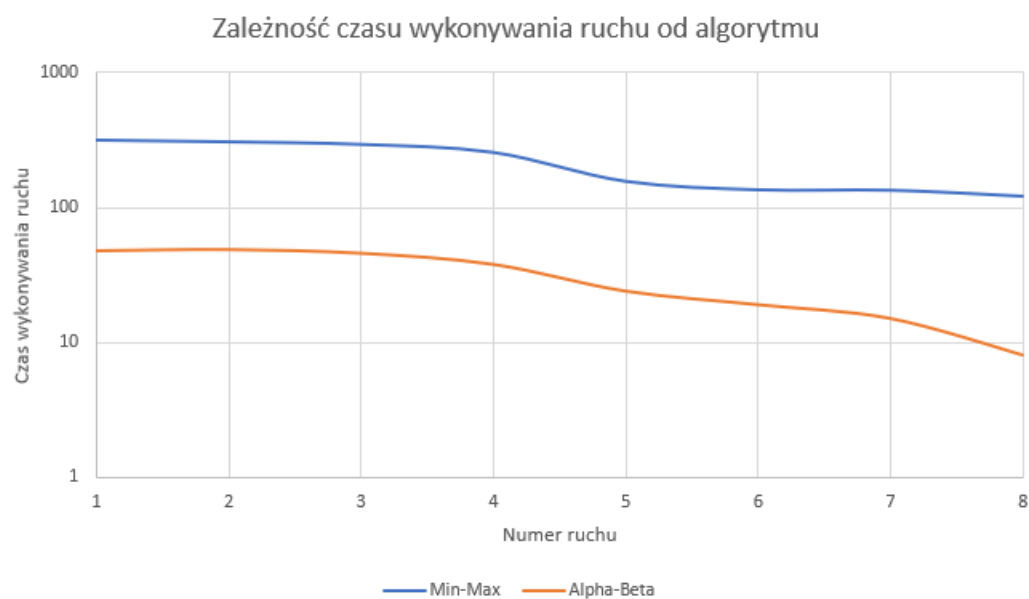
Zbadano czas wykonywania ruchów w pojedynkach obu algorytmów dla zadanych głębokości. Wykorzystano funnkcję oceny Changed Point - Blocking oraz brak losowości pierwszego ruchu.

2.5.1 Głębokość na poziomie 3



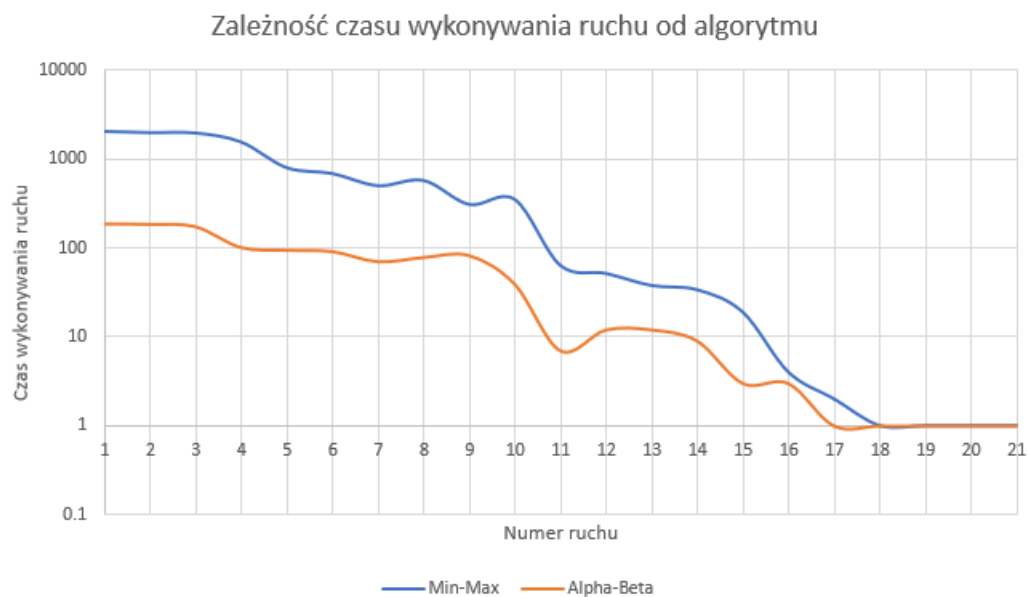
Zwyciężył Alpha-Beta

2.5.2 Głębokość na poziomie 4

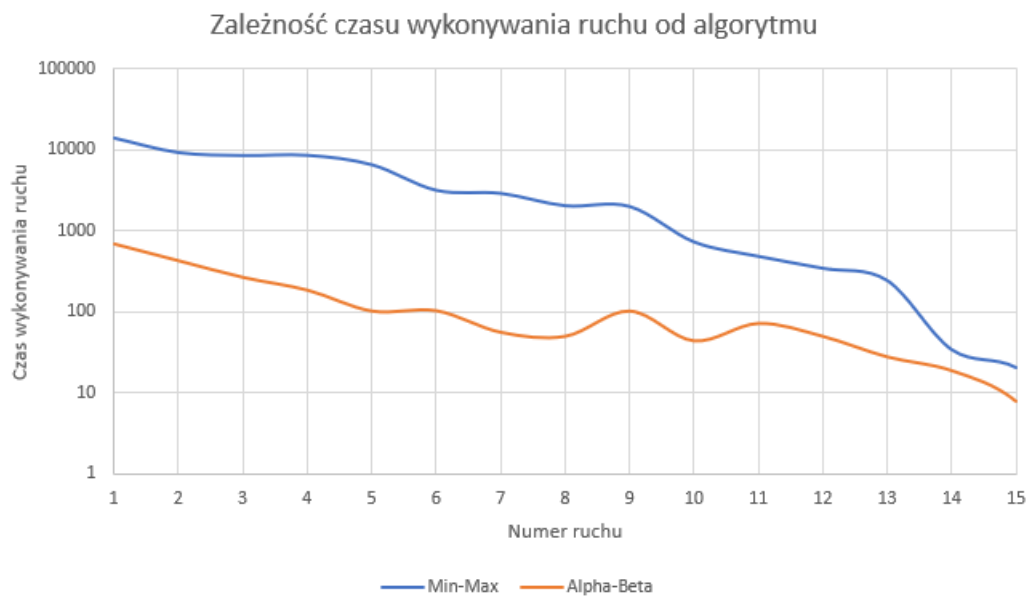


Zwyciężył Alpha-Beta

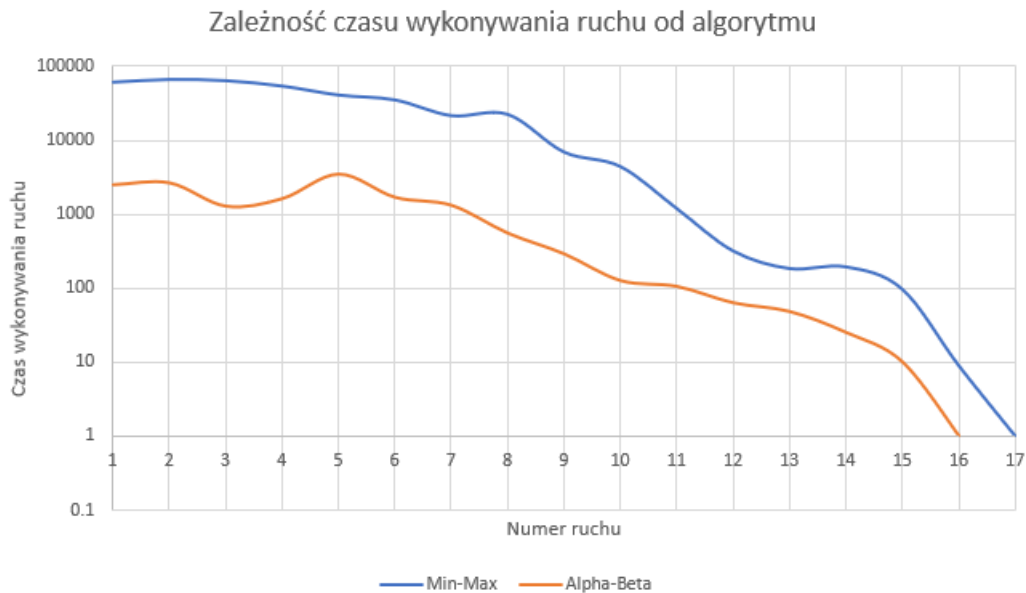
2.5.3 Głębokość na poziomie 5



2.5.4 Głębokość na poziomie 6



2.5.5 Głębokość na poziomie 7



Zwyciężył Min-Max

1 Wnioski

Badanie wykazało przewagę Alfa-Beta nad Min-Maxem. Znacznie częściej wygrywa Alfa-Beta, niż Min-Max. Do tego ten pierwszy dla dowolnej głębokości gwarantuje lepsze czasy wyboru pola.

Powodem tego oczywiście jest charakterystyczna cecha Alfa-Bety, który dokonuje cięć i wyklucza poddrzewa, do których nie warto wchodzić w poszukiwaniu lepszego rozwiązania.

Min-Max zdaje się być skuteczniejszy dla większych głębokości, tych rzędu 6-7. Dla prostszych i mniej skomplikowanych sprawdzeń niezawodny jest Alfa-Beta.

3 Podsumowanie

Czas znajdowania kolejnych ruchów jest silnie związany z wyborem heurystyki. Dla heurystyk z kategorii Whole Board czas jest znacznie dłuższy niż dla pozostałych. Powodem tego jest inny sposób przeliczania wartości planszy; heurystyki Whole Board analizują całą planszę (a konkretniej - te miejsca, w których są już wykonane jakieś ruchy), podczas gdy heurystyki z kategorii Changed Point analizują zawsze tylko jedno miejsce - to najpóźniej zmodyfikowane na planszy.

Heurystyki, w których ocena planszy bierze pod uwagę blokowanie sprawiają, że rozgrywka trwa dłużej. Może to świadczyć o tym, że taka heurystyka lepiej oddaje rzeczywistość, jeśli chodzi o odpowiednią ocenę rozgrywki.

Wraz ze wzrostem głębokości poszukiwań, rośnie liczba ruchów gracza do momentu zwycięstwa (lub zakończenia) gry. Jest to związane z lepszą skutecznością algorytmu dla większych głębokości.