

Algorytm Mini-max

Marcin Połosak

Listopad 2023

1 Opis algorytmu

Algorytm min-max opiera się na założeniu, że gracz podczas swojego ruchu będzie chciał uzyskać najlepszy możliwy rezultat podczas gdy jego przeciwnik będzie wybierał dla niego warianty najgorsze.

Najpewniejszą opcją jest zawsze sprawdzenie rozwiązania dla każdej logicznie możliwej drogi. Wówczas możemy założyć, że ocenę uzyskanego ruchu można przeprowadzić następująco:

1 - wygrywasz
0 - jest remis
-1 - wygrywa przeciwnik

W przypadku prostych gier o małej ilości możliwości realne jest sprawdzanie wszystkich możliwości. W przypadku rozgywek z większą ilością możliwych kombinacji policzenie wszelkich rozwiązań przekraczało by akceptowalny czas. Wówczas należy ograniczyć przeszukiwanie do określonej głębokości i określić, która pozycja zdaje się być najlepsza.

Określenie odpowiedniej heurystyki stanowi najcięższe zadanie w implementacji poprawnie działającego algorytmu.

W przypadku gry w kółko i krzyżyk dobrym punktem wyjścia jest zliczanie kombinacji dwóch oraz trzech, w przypadku gry na polu większym niż 3x3, takich samych symbolów koło siebie. Co ważne, istotne jest również, z ilu stron ograniczona jest dana grupa. O ile taka zamknięta z dwóch stron przez przeciwnika nie stanowi podstawy do uznania przewagi, o tyle kombinacja 2 (lub 3 w przypadku gry na większych polach) takich samych symbolów otoczonych dwoma pustymi polami stanowi pozycję, której przeciwnik nie jest już w stanie zablokować.

1.1 $\alpha - \beta$

Kluczowym usprawnieniem algorytmu, które znacząco skraca czas oczekiwania na wynik jest na pewno wariant $\alpha - \beta$. Zakłada on, że jeśli w celu dostania się do jakiejś gałęzi drzewa przeciwnik musiałby wybrać dla niego opcję gorszą od aktualnie najlepszej, to nie ma powodu sprawdzania tej gałęzi. Dzięki tej implementacji znacząco ograniczyłem czas przeszukiwania algorytmu co umożliwiło korzystanie z algorytmu z większą głębokością przeszukiwań.

1.2 Korekta funkcji oceny

Kolejnym usprawnieniem, którego konieczność zauważyłem w trakcie pracy z algorytmem jest poprawa funkcji oceny stanów terminalnych. Wersja podstawowa zakładała, że każda wygrana jest tak samo warta. W przypadku, gdy algorytm był pewny, że niezależnie który z kilku ruchów wybierze wygra, wybierał pierwszy, który zauważył. Zdecydowałem, że ważne jest, żeby algorytm wybierał zwycięstwo szybsze, to jest takie, które nastąpi po możliwie najmniejszej liczbie ruchów.

1.3 Możliwe usprawnienia

Istnieje wiele innych możliwych usprawnień algorytmu. Można chociażby decydować o pomijaniu ruchów nieznaczających czyli takich, które dzieją się w drugim końcu planszy.

W ramach implementacji całej gry można również zdecydować się na zwiększanie głębokości przeszukiwania wraz z malejącą liczbą wolnych pól.

2 Eksperymenty

W ramach eksperymentów porównam sprawność algorytmów o różnych głębokościach przeszukiwań.

Elementem faworyzującym w przypadku gry w kółko i krzyżyk jest pierwszeństwo w wykonywaniu ruchów, dlatego postaram się pokazać, że w przypadku odpowiedniej różnicy w głębokości przeszukiwania nawet algorytm wykonujący ruch jako drugi jest w stanie wygrać.

Dodatkowo zaprezentuję wpływ głębokości przeszukiwań na czas obliczeń w kolejnych iteracjach.

3 Rezultaty

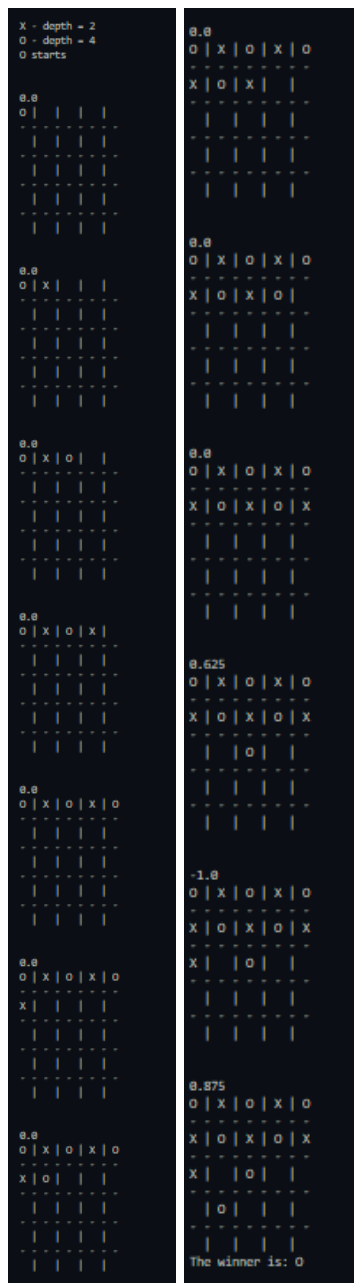
3.1 3x3

<pre> X - depth = 2 O - depth = 3 O starts </pre> <pre> 0.5 O - - - - - - - - </pre> <pre> 0.0 O X - - - - - - - - </pre> <pre> 0.8 O X - - - - O - - - - </pre> <pre> 0.0 O X - - - - O - - - - X </pre> <pre> 1.0 O X - - - - O O - - - - X </pre> <pre> -1.0 O X X - - - - O O - - - - X </pre> <pre> 1.0 O X X - - - - O O O - - - - X The winner is: O </pre>	<pre> X - depth = 2 O - depth = 3 X starts </pre> <pre> 0.0 X - - - - - - - - </pre> <pre> 0.0 X O - - - - - - - - </pre> <pre> 0.0 X O X - - - - - - - - </pre> <pre> 0.5 X O X - - - - O - - - - </pre> <pre> 0.0 X O X - - - - O - - - - X </pre> <pre> 0.0 X O X - - - - O O - - - - X </pre> <pre> -0.8 X O X - - - - O O X - - - - X </pre> <pre> 0.0 X O X - - - - O O X - - - - X O </pre> <pre> 0.0 X O X - - - - O O X - - - - X X O There is a draw </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

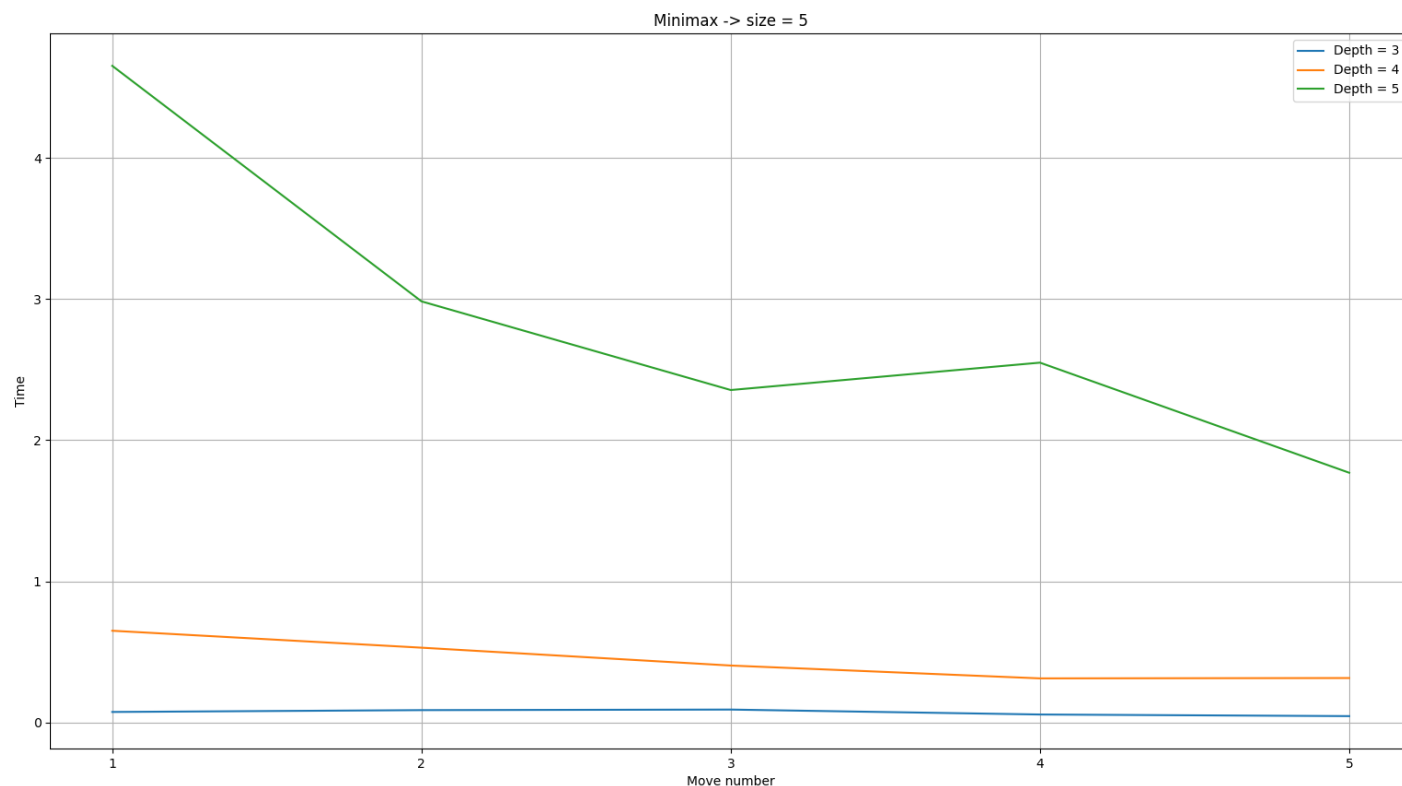
Warto zauważyć wpływ rozpoczynania na rezultat. Co ważne, w opu przypadkach algorytm dokładniejszy nie przegrał.

<p>X - depth = 1 O - depth = 5 X starts</p> <pre> 0.0 X - - - - - - - - - - </pre> <p>0.5</p> <pre> X - - - - - O - - - - - </pre> <p>0.5</p> <pre> X X - - - - - O - - - - - </pre> <p>0.0</p> <pre> X X O - - - - - O - - - - - </pre> <p>0.5</p> <pre> X X O - - - - - X O - - - - - </pre> <p>1.0</p> <pre> X X O - - - - - X O - - - - - O </pre> <p>The winner is: O</p>	<p>X - depth = 1 O - depth = 9 O starts</p> <pre> 0.0 O - - - - - - - - - - </pre> <p>0.0</p> <pre> O X - - - - - - - - - - </pre> <p>1.0</p> <pre> O X - - - - - O - - - - - </pre> <p>0.0</p> <pre> O X - - - - - O - - - - - X </pre> <p>1.0</p> <pre> O X - - - - - O O - - - - - X </pre> <p>-0.8</p> <pre> O X X - - - - - O O - - - - - X </pre> <p>1.0</p> <pre> O X X - - - - - O O O - - - - - X </pre> <p>The winner is: O</p>	<p>X - depth = 4 O - depth = 9 O starts</p> <pre> 0.0 O - - - - - - - - - - </pre> <p>0.0</p> <pre> O X - - - - - - - - - - </pre> <p>1.0</p> <pre> O X - - - - - O - - - - - </pre> <p>-1.0</p> <pre> O X X - - - - - O - - - - - </pre> <p>1.0</p> <pre> O X X - - - - - O O - - - - - </pre> <p>-1.0</p> <pre> O X X - - - - - O O X - - - - - </pre> <p>1.0</p> <pre> O X X - - - - - O O X - - - - - O </pre> <p>The winner is: O</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.2 5x5



Jak widać, w przypadku większej planszy różnice w jakości algorytmów były jeszcze bardziej widoczne. Algorytm o większej głębokości przeszukiwań był w stanie założyć pułapkę, której gorszy wariant nie mógł zauważyć.



3.3 Czas poszukiwania rozwiązań

Wykres ten pokazuje silną zależność czasu oczekiwania w zależności od głębokości przeszukiwania. O ile dla małych wartości nie jest to problem, o tyle przy większych wartościach głębokości przeszukiwania algorytm potrzebuje dłuższego czasu.

Wynika to z faktu, że liczba możliwości w grze kółko i krzyżyk jest ogromna.

dla rozmiaru 3x3 — 362,880

4 Wnioski

Udało się zaimplementować algorytm zdolny do przeszukiwania grafu w celu wybierania poprawnych ruchów w grze w kółko i krzyżyk.

Można zauważyć, że wraz ze wzrostem głębokości przeszukiwań zwiększa się jakość rozwiązań. Im większa głębokość tym algorytm jest w stanie znaleźć bardziej skomplikowane rozwiązania, wymagające większej ilości ruchów.

Dzięki sprawdzeniu algorytmów na większych planszach można było zauważyć jak algorytm poszukuje rozwiązań najlepszych. Jest w stanie również udowodnić swoją większość wygrywając z graczem, którego faworyzuje fakt, że to on rozpoczynał.

Ograniczeniem jest oczywiście moc obliczeniowa algorytmów. Dlatego właśnie konieczne jest ulepszanie algorytmu eliminując sprawdzanie rozwiązań nielogicznych. Jednym z takich rozwiązań jest właśnie użycie wariantu $\alpha - \beta$.

Największy problem zadania to określenie odpowiedniej heurystyki. To właśnie ona jest kluczowa w przypadku gier dwuosobowych. Liczba możliwych kombinacji i wariantów często uniemożliwia badanie wszelkich możliwych założeń, które świadczą o przewadze jednego gracza nad drugim.